

**StorageTek Storage Archive Manager and
StorageTek QFS Software**

File System Recovery Guide

Release 5.4

E42065-02

March 2015

Copyright © 2011, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Robert Craig Johnson

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	v
Documentation Accessibility	v
Prerequisites for Using this Document	v
Conventions	v
Available Documentation	vi
 1 Introduction	
Failure and Recovery Scenarios.....	1-1
Recommended Preparations	1-2
 2 Stabilizing the Situation	
Stopping Archiving and Recycling Processes	2-1
Stop Archiving.....	2-1
Stop Recycling	2-3
Preserving Unarchived Data	2-3
Back Up Unarchived Files.....	2-3
Preserving Configuration and State Information	2-4
Save the SAM-QFS Configuration	2-4
 3 Restoring the SAM-QFS Configuration	
Restoring the Configuration from Backup Copies and/or SAMreports.....	3-1
Restore the Configuration from Backup Files or SAMreports	3-1
Restoring the Configuration Without Backup Information	3-4
 4 Recovering File Systems	
Recreating the File System	4-1
Recreate the File System Using Backup Configuration Files and the sammkfs Command	4-1
Restoring Directories and Files	4-3
Restoring Files and Directories from a samfsdump (qfsdump) Recovery Point File.....	4-4
Restore the Lost File System from a Recovery Point File.....	4-4
Restage Archived Files If Required	4-5
Identify Damaged Files and Locate Replacement Copies.....	4-5
Identify Files that Were Archived After the Recovery Point Was Created	4-7
Restore the Damaged and/or Missing Files	4-9

Restoring Files and Directories from Archival Media without a Recovery Point File.....	4-9
--	-----

5 Recovering Lost and Damaged Files

Recovering Files Using a Recovery Point File	5-1
Recovering Files Using Archiver Log Entries	5-2
Restore Lost and Damaged Regular Files.....	5-3
Restore Lost and Damaged Segmented Files	5-4
Restore Lost and Damaged Volume Overflow Files.....	5-7
Recovering Damaged Archive Copies.....	5-10

6 Finishing Up

Restoring Archiving File Systems to Normal Operation	6-1
Enable Archiving.....	6-1
Enable Recycling	6-2
Preserving the New Configuration Information.....	6-3
Save the Newly Restored SAM-QFS Configuration	6-3

A Understanding the Archiver Log

B Glossary of Equipment Types

Recommended Equipment and Media Types	B-1
Other Equipment and Media Types	B-2

C Product Accessibility Features

Glossary

Preface

This document addresses the needs of system administrators, storage and network administrators, and service engineers who may have to recover damaged or lost SAM-QFS files, file systems, and archiving solutions.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit
<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit
<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Prerequisites for Using this Document

This document assumes that you are already familiar with Oracle Solaris operating system, storage, and network administration. Please refer to the Solaris documentation and man pages and to storage hardware documentation for information on relevant tasks, commands, and procedures.

Conventions

The following textual conventions are used in this document:

- *Italic* type represents book titles and emphasis.
- `Monospace` type represents commands and text displayed in a terminal window and the contents of configuration files, shell scripts, and source code files.
- **Monospace bold** type represents user inputs and significant changes to commandline output, terminal displays, or file contents. It may also be used to emphasize particularly relevant parts of a file or display.
- **Monospace bold oblique** type represents variable inputs and outputs in a terminal display or file.
- *Monospace oblique* type represents other variables in a terminal display or file.
- ... (three-dot ellipsis marks) represent file contents or command output that is not relevant to the example and has thus been omitted for brevity or clarity.

- \ (a backslash) at the end of a line in examples escapes the line break so that the following line is part of the same command.
- [-] (brackets surrounding values separated by a hyphen) delimit value ranges.
- [] (brackets) in command syntax descriptions indicate optional parameters.
- root@solaris:~# and [hostname]:root@solaris:~# represent Solaris command shell prompts.
- [root@Linux ~]# represents Linux command shell prompts.

Available Documentation

The *StorageTek Storage Archive Manager and StorageTek QFS Software File System Recovery Guide* is part of the multivolume *SAM-QFS Customer Documentation Library*, available from docs.oracle.com.

For Solaris operating system documentation, see the *Oracle Solaris 11.1 Information Library*, also available from docs.oracle.com.

Introduction

This document outlines the steps that you should take in order to recover files and file systems that have been lost or corrupted due to hardware failure, misconfiguration, human error, or the physical destruction of facilities and equipment. Properly configured SAM-QFS file systems are extremely robust. But the steps that you need to take during recovery—and your probability of success—depend on your degree of preparedness. So this introduction begins with an overview of the recovery process. Then it moves to a review of the data- and file system-protection measures that Oracle recommends. Finally it outlines the steps open to you based on the preparations that you have made and the resources that you currently have available.

Failure and Recovery Scenarios

The scope of a file-system failure and the nature of the required recovery actions depends on the nature of the underlying problem. For example:

- If the server host fails, the SAM-QFS software and file-system configurations may be lost, leaving file system data and metadata intact but inaccessible until the configuration information is restored.

Once the underlying hardware problem has been addressed and the operating system has been restored, the SAM-QFS configurations can be restored by reinstalling the software and installing backup copies of the configuration files and any custom scripts. In this situation, follow the procedures outlined in [Chapter 3, "Restoring the SAM-QFS Configuration"](#).

- If an administrator inadvertently deletes or misconfigures one or more configuration files, library catalogs, scripts, or `crontab` entries, access to one or more file systems may be lost along with some or all software functionality.

The SAM-QFS configurations can be restored by reinstalling backup copies of the affected configuration files and/or scripts. Follow the procedures outlined in [Chapter 3, "Restoring the SAM-QFS Configuration"](#).

- If a disk or RAID group that provides the disk cache for the data in a standalone (non-archiving) file-system fails, all files in the disk cache are lost.

Once the hardware problem has been addressed, lost files can be restored from QFS backup copies. See ["Recovering Files Using a Recovery Point File"](#) on page 5-1.

- If a disk or RAID group that provides the disk cache for the data in an archiving file-system fails, all files in the disk cache are lost.

Once the hardware problem has been addressed, lost files can be restored from archived copies or SAM-QFS backup files. See ["Recovering Files Using a Recovery](#)

[Point File](#)" on page 5-1 and ["Recovering Files Using Archiver Log Entries"](#) on page 5-2.

- If the disks that store file system metadata fail, the file system is lost and the data is no longer readily accessible.

Once the hardware problem has been addressed, metadata can be restored from backup files. If the metadata for an archiving file system was not backed up, it can be reconstructed from backup copies of the archiver log file. See [Chapter 5, "Recovering Lost and Damaged Files"](#).

- If an administrator inadvertently formats the disk partitions that host a SAM-QFS file system or issues the `sammkfs` command against existing SAM-QFS partitions, all files and metadata are lost.

Metadata can be restored from backup files or reconstructed from the archiver log of an archiving file system. Data can be restored from archival media or from a backup file. See [Chapter 5, "Recovering Lost and Damaged Files"](#).

Recommended Preparations

In the *StorageTek Storage Archive Manager and StorageTek QFS Installation and Configuration Guide*, Oracle recommends that you include the following configuration, file-system, and data backup steps in your initial configuration:

- Store critical data in SAM-QFS archiving file systems and archive at least two copies of the file data, at least one of which should be on removable media, such as magnetic tape. If possible, configure disk archives on independent file systems that do not share physical devices with the disk cache of the archiving file-system.
- Store file-system metadata on highly redundant, mirrored storage.
- Regularly backup SAM-QFS file systems with recovery point files. A recovery point file stores file-system metadata and, optionally, data, so that files or entire file systems can be restored.

If you have the Storage Archive Manager software installed, you create recovery point files by running the `samfsdump` command. If you have only the QFS file system software, you use the `qfsdump` command. Using either command on its own backs up the metadata. Using either command with the `-U` option backs up data as well as metadata. This option is mainly useful for protecting file systems that are not archived to removable media. You can run the dump commands from the command line or from the SAM-QFS Manager graphical user interface.

- Configure the host to automatically save SAM-QFS metadata recovery point files, either by using entries in the `crontab` file or by using the scheduling feature of the SAM-QFS Manager.
- Configure the host to automatically save SAM-QFS archiver log information. Create entries in the system `crontab` file.

For each file that is archived with the Storage Archive Manager software, the archiver log file records the file's name and location (path) within the file system, the name of archive (`tar`) files that hold copies, the removable media volumes that hold the archive files, and the positions of the archive files on the media.

- Select a secure storage location for all SAM-QFS recovery information, including backed up configuration files, recovery points, and archiver logs. Select an independent file system that will mount on the SAM-QFS file system host but does not share any physical devices with the archiving file system.

Do not store disaster recovery resources in the file system that they are meant to protect. Do not use logical devices, such as partitions or LUNs, that reside on physical devices that also host the archiving file-system.

Stabilizing the Situation

Whenever you are faced with recovering from a significant file-system failure or potential data loss, your first step should be to stabilize the affected systems, minimize chances for further losses, and preserve diagnostic information, where possible. When SAM-QFS hosts remain viable and normal operations continue in an abnormal situation, additional damage is inadvertently done. This chapter outlines some basic steps that help to avoid this:

- [Stopping Archiving and Recycling Processes](#) when archiving file systems are being restored.
- [Preserving Unarchived Data](#)
- [Preserving Configuration and State Information.](#)

Stopping Archiving and Recycling Processes

Whenever you are restoring an archiving file system and have to recover significant numbers of files, your first step should be to stop the archiving and recycling processes, either for the file system or for the entire system. You want to stabilize and isolate the archive until you have assessed the situation and, ideally, restored everything to normal. Otherwise, ongoing archiving and recycling operations can, in some situations, make matters worse by propagating corrupted files and/or by recycling remaining archive copies that contain valid data.

So, whenever possible, take the precautions listed below:

- [Stop Archiving](#)
- [Stop Recycling.](#)

Once recovery operations are complete, you can reverse the changes below and restore normal file system behavior.

Stop Archiving

1. Log in to the file-system metadata server as root.

```
root@solaris:~#
```

2. Open the `/etc/opt/SUNWsamfs/archiver.cmd` file in a text editor, and scroll down to the first `fs` (file-system) directive.

In the example, we use the `vi` editor:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# Configuration file for SAM-QFS archiving file systems
#-----
```

```
# General Directives
archivemeta = off
examine = noscan
#-----
# Archive Set Assignments
fs = samqfs1
logfile = /var/adm/samqfs1.archive.log
all .
    1 -norelease 15m
    2 -norelease 15m
fs = samqfs2
logfile = /var/adm/samqfs2.archive.log
all .
...
```

3. To stop archiving on all file systems, add a `wait` directive to file immediately before the first `fs` directive. Save the file, and close the editor.

In the example, we insert the `wait` directive just before the directive for the `samqfs1` file system, where it will apply to all file systems configured for archiving:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#-----
# Archive Set Assignments
wait
fs = samqfs1
logfile = /var/adm/samqfs1.archive.log
all .
    1 -norelease 15m
    2 -norelease 15m
    3 -norelease 15m
fs = samqfs2
...
:wq
root@solaris:~#
```

4. If you need to stop archiving on only one file system, add a `wait` directive to file immediately after the corresponding `fs` directive. Save the file, and close the editor.

In the example, we stop archiving activity on the `samqfs1` file system:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#-----
# Archive Set Assignments
fs = samqfs1
wait
logfile = /var/adm/samqfs1.archive.log
all .
    1 -norelease 15m
    2 -norelease 15m
    3 -norelease 15m
fs = samqfs2
...
:wq
root@solaris:~#
```

5. Next, [Stop Recycling](#).

Stop Recycling

1. Log in to the file-system metadata server as root.

```
root@solaris:~#
```

2. Open the `/etc/opt/SUNWsamfs/recycler.cmd` file in a text editor.

In the example, we use the vi editor:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for SAM-QFS archiving file systems
#-----
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
library1 -hwm 95 -mingain 60
```

3. Add the `-ignore` parameter to the recycling directive for each media library. Then save the file, and close the editor.

In the example, we have only one library in the SAM-QFS configuration, `library1`:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for SAM-QFS archiving file systems
#-----
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
library1 -hwm 95 -mingain 60 -ignore
:wq
root@solaris:~#
```

4. If you are recovering from loss or damage to one more archiving file systems, [Back Up Unarchived Files](#) before proceeding.
5. If you are recovering from a server problem or from loss or damage to file systems, [Save the SAM-QFS Configuration](#) before proceeding.
6. If you need to restore directories and files, decide whether you need to [Save the SAM-QFS Configuration](#) or go directly to [Chapter 5, "Recovering Lost and Damaged Files"](#).

Preserving Unarchived Data

Unarchived files may remain in the disk cache of a damaged archiving file system. No copies of these files exist in the archive. So, if we can, we need to back them up to a recovery point file before proceeding. Proceed as follows:

Back Up Unarchived Files

1. If possible, log in to the file-system metadata server as root.

```
root@solaris:~#
```

2. Select a safe location for the recovery point will be stored.

In the example, we create a subdirectory, `unarchived/`, under the directory that we created for recovery points when we configured file systems (see the *StorageTek Storage Archive Manager and StorageTek QFS Installation and Configuration Guide*).

```
root@solaris:~# mkdir /zfs1/samqfs_recovery/unarchived/
root@solaris:~#
```

3. Change to the file system's root directory.

In the example, we change to the mount-point directory `/samqfs1`:

```
root@solaris:~# cd /samqfs1
root@solaris:~#
```

4. Backup any unarchived files that remain in the disk cache. Use the command `samfsdump -u -f recovery-point`, where *recovery-point* is the path and file name of the output file.

The `-u` option causes the `samfsdump` command to backup any data files that have not been archived. In the example, we save the recovery point file 20140404 to the remote directory `/zfs1/samqfs_recovery/unarchived/`:

```
root@solaris:~# samfsdump -u -f /zfs1/samqfs_recovery/unarchived/20140404
root@solaris:~#
```

5. If you are recovering from a server problem or from loss or damage to file systems, [Save the SAM-QFS Configuration](#) before proceeding.
6. If you need to restore directories and files, decide whether you need to [Save the SAM-QFS Configuration](#) or go directly to [Chapter 5, "Recovering Lost and Damaged Files"](#).

Preserving Configuration and State Information

Even when you have safely stored backup copies of all configuration files and scripts needed for restoring the SAM-QFS software and file-system, it pays to preserve the current state of a failed system, if you can. Surviving configuration files and scripts may contain changes that were implemented since the complete configuration was last backed up. This can mean the difference between restoring the system to its almost its exact pre-failure state and merely getting close. Log and trace files contain information that helps restore files and clarifies the causes of failures. For this reason, you should preserve whatever remains, before you do anything else.

Save the SAM-QFS Configuration

1. If possible, log in to the file-system metadata server as `root`.

```
root@solaris:~#
```

2. If you can run the `samexplorer` command create a SAMreport and save it in the directory that holds your backup configuration information. Use the command `samexplorer path/hostname.YYYYMMDD.hmmz.tar.gz`, where *path* is the path to the chosen directory, *hostname* is the name of the SAM-QFS file system host, and *YYYYMMDD.hmmz* is a date and time stamp.

The default file name is `/tmp/SAMreport.hostname.YYYYMMDD.hmmz.tar.gz`. In the example, we already have a directory for saving SAMreports, `/zfs1/sam_config/`. So we create the report in this directory:

```
root@solaris:~# samexplorer
/zfs1/sam_config/explorer/server1.20140430.1659MST.tar.gz
Report name:
/zfs1/sam_config/explorer/samhost1.20140430.1659MST.tar.gz
Lines per file: 1000
Output format: tar.gz (default) Use -u for unarchived/uncompressed.

Please wait.....
```

```
Please wait.....
Please wait.....
```

The following files should now be ftp'ed to your support provider as ftp type binary.

```
/zfs1/sam_config/explorer/samhost1.20140430.1659MST.tar.gz
```

3. Copy as many of the SAM-QFS configuration files as you can to another file system. These include the following:

```
/etc/opt/SUNWsamfs/
  mcf
  archiver.cmd
  defaults.conf
  diskvols.conf
  hosts.family-set-name
  hosts.family-set-name.local
  preview.cmd
  recycler.cmd
  releaser.cmd
  rft.cmd
  samfs.cmd
  stager.cmd
  inquiry.conf
  samremote                # SAM-Remote server configuration file
  family-set-name          # SAM-Remote client configuration file
  network-attached-library # Parameters file
  scripts/*                # Back up all locally modified files
/var/opt/SUNWsamfs/
```

4. Back up all surviving library catalog data, including that maintained by the historian, if possible. For each catalog, use the command
`/opt/SUNWsamfs/sbin/dump_cat -V catalog-file`, where *catalog-file* is the path and name of the catalog file. Redirect the output to *dump-file*, in a new location.

We will use the output of the `dump_cat` file to rebuild the catalogs on a replacement system, using the command `build_cat`. In the example, we dump the catalog data for `library1` to the file `library1cat.dump` in a directory on the independent NFS-mounted file system `zfs1`:

```
root@solaris:~# dump_cat -V /var/opt/SUNWsamfs/catalog/library1cat > \
/zfs1/sam_config/20140513/catalogs/library1cat.dump
```

5. Copy system configuration files that were modified during SAM-QFS installation and configuration. These may include:

```
/etc/
  syslog.conf
  system
  vfstab
/kernel/drv/
  sgen.conf
  samst.conf
  samrd.conf
  sd.conf
  ssd.conf
  st.conf
/usr/kernel/drv/dst.conf
```

6. Copy any custom shell scripts and `crontab` entries that you created as part of the SAM-QFS configuration to the selected subdirectory.
For example, if you created a `crontab` entry to manage creation of recovery points, you would save a copy now.
7. Record the revision level of the currently installed software, including Oracle SAM-QFS, Solaris, and Solaris Cluster (if applicable), and save a copy of the information in a `readme` file in the chosen subdirectory.
8. In the chosen subdirectory, save copies of downloaded Oracle SAM-QFS, Solaris, and Solaris Cluster packages so that you can restore the software quickly, should it become necessary.
9. If you are recovering from the loss of a SAM-QFS server host, go to [Chapter 3, "Restoring the SAM-QFS Configuration"](#).
10. If you need to restore one or more SAM-QFS file systems, go to [Chapter 4, "Recovering File Systems"](#).
11. If you need to restore directories and files, go to [Chapter 5, "Recovering Lost and Damaged Files"](#).

Restoring the SAM-QFS Configuration

This chapter outlines the process of recovering the SAM-QFS software and file-system configuration in the event that it is lost or corrupted, either in part or in its entirety. If the server host fails, the SAM-QFS software and file-system configurations may be lost, leaving file system data and metadata intact but inaccessible until the configuration information is restored. Success in this situation depends on your ability to salvage information from any files and directories that remain and on the thoroughness of your disaster preparations:

- [Restoring the Configuration from Backup Copies and/or SAMreports](#)
- [Restoring the Configuration Without Backup Information.](#)

Restoring the Configuration from Backup Copies and/or SAMreports

If you followed the procedures recommended in the *StorageTek Storage Archive Manager and StorageTek QFS Installation and Configuration Guide*, you can recover SAM-QFS software and file-system configurations using the procedure below.

Restore the Configuration from Backup Files or SAMreports

1. If you are restoring the configuration following a server host failure, resolve hardware issues, re-install operating systems and software as needed.
2. If a current backup copy of the root file system exists, restore the root file system and stop here.
3. Otherwise, log in to the file-system server host as `root`.

```
root@solaris:~#
```

4. Mount any required file systems, including file systems that store backup SAM-QFS configuration files and any file systems that hold disk-based archive copies of data files.

In the example, we have been maintaining copies of the SAM-QFS server's Solaris configuration files in the subdirectory `sam_config` on the independent file system `zfs1`. So we create a mount point, mount `zfs1`, restore the `vfstab` file from the most recent copy, create mount points as needed, and then mount the necessary file systems:

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.back
root@solaris:~# mkdir /zfs1
root@solaris:~# mount -F zfs /net/remote.example.com/zfs1/ /zfs1
root@solaris:~# cp /zfs1/sam_config/20140127/etc/vfstab /etc/vfstab
root@solaris:~# mkdir /diskvols
```

```
root@solaris:~# mkdir /diskvols/DISKVOL1
root@solaris:~# mkdir /diskvols/DISKVOL2
...
root@solaris:~# mount /diskvols/DISKVOL1
root@solaris:~# mount /diskvols/DISKVOL2
...
root@solaris:~#
```

5. If backup copies of the SAM-QFS configuration files are available, locate the most recent that is dated prior to the loss of the configuration.

In the example, we have been maintaining copies of the SAM-QFS configuration files in the subdirectory `sam_config`, on the independent file system `/zfs1`. So the latest files are easy to find:

```
root@solaris:~# ls /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/
archiver.cmd      defaults.conf      mcf                recycler.cmd       stager.cmd
cfg_backups       diskvols.conf      mgmt_sched.conf    releaser.cmd       startup
csn               inquiry.conf       notify.cmd         scripts            verifyd.cmd
root@solaris:~# ls /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/scripts
archiver.sh      log_rotate.sh      nrecycler.sh       recycler.sh        save_core.sh
sendtrap         ssi.sh
root@solaris:~# ls /zfs1/sam_config/explorer/
server1.20140430.1659MST.tar.gz  server1.20140114.0905MST.tar.gz
server1.20110714.1000MST.tar.gz
```

6. If SAMreports were generated before the loss of the SAM-QFS configuration, locate the most recent.

In the example, we have

`/zfs1/sam_config/explorer/server1.20140127.1659MST.tar.gz`.

7. If affected file systems are currently mounted, unmount them.
8. For each missing configuration file, copy an available backup file to the required location on the server that you are restoring.

In the example, we restore all of the SAM-QFS configuration files and scripts from backup copies.

```
root@solaris:~# cp /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/* \
/etc/opt/SUNWsamfs/
root@solaris:~# cp /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/scripts/* \
/etc/opt/SUNWsamfs/scripts/
root@solaris:~# cp /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/startup/* \
/etc/opt/SUNWsamfs/startup/
root@solaris:~# cp /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/cfg_backups/* \
/etc/opt/SUNWsamfs/cfg_backups/
root@solaris:~# cp /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/csn/* \
/etc/opt/SUNWsamfs/csn/
```

9. If backup copies of the configuration files are not available, recreate them using the information contained in the most recent available SAMreport. Copy the content from the report, paste it into a text editor, and save it to the file and path indicated in the report.

SAMreport files contain the full text of the SAM-QFS files as they were at the time the report was created. They also list the directory where the file was located.

In the example, we open the file `server1.20140127.SAMreport` in the `vi` editor and search for the regular-expression pattern `\etc\opt\SUNWsamfs\mcf`. We find the following information on the `mcf` file:

```

root@solaris:~# vi /zfs1/sam_config/explorer/server1.20140127.SAMreport
...
----- /etc/opt/SUNWsamfs/mcf -----
server1# /bin/ls -l /etc/opt/SUNWsamfs/mcf
-rw-r--r--  1 root    root      1789 Feb  4 09:22 /etc/opt/SUNWsamfs/mcf

# Equipment      Equipment  Equipment  Family    Device    Additional
# Identifier      Ordinal    Type       Set       State     Parameters
#-----
samqfs1          100        ms         samqfs1   on
/dev/dsk/c1t3d0s3 101        md         samqfs1   on
/dev/dsk/c1t4d0s5 102        md         samqfs1   on
~
/\etc\opt\SUNWsamfs\mcf

```

10. Restore the library catalogs from the dump-file data that you saved during the procedure ["Save the SAM-QFS Configuration"](#) on page 2-4. For each catalog, use the command `/opt/SUNWsamfs/sbin/build_cat catalog-dump-file catalog-file`, where *catalog-dump-file* is the path and name of the file that you created with the `dump_cat` command and *catalog-file* is the path and name of the restored catalog file.

In the example, we rebuild the catalog for library1 using the data in the file `/zfs1/sam_config/20140513/catalogs/library1cat.dump`:

```

root@solaris:~# build_cat /zfs1/sam_config/20140513/catalogs/library1cat.dump \
/var/opt/SUNWsamfs/catalog/library1cat

```

11. If you are recovering a system following hardware failure, go to [Chapter 4, "Recovering File Systems"](#).
12. If you are replacing one or more configuration files that were inadvertently deleted or incorrectly edited and no hardware or file system changes have occurred, check the configuration files for errors by running the `sam-fsd` command.

The `sam-fsd` is an initialization command that reads SAM-QFS configuration files. It will stop if it encounters an error:

```

root@solaris:~# sam-fsd

```

13. If the `sam-fsd` command finds an error in the `mcf` file, edit the file to correct the error and recheck as described in the preceding step.

In the example below, `sam-fsd` reports an unspecified problem with a device:

```

root@solaris:~# sam-fsd
Problem in mcf file /etc/opt/SUNWsamfs/mcf for filesystem samqfs1
sam-fsd: Problem with file system devices.

```

14. If the `sam-fsd` command runs without error, the configuration files are correct. Proceed to the next step.

The example is a partial listing of error-free output:

```

root@solaris:~# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
               cust err fatal ipc misc proc date
               size  10M age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
               cust err fatal ipc misc proc date module

```

```
size      10M  age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
cust err fatal ipc misc proc date module
size      10M  age 0
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
```

15. Tell the SAM-QFS software to read the `mcf` file and reconfigure itself accordingly. Use the command `samd config`.

```
root@solaris:~# samd config
```

16. If the `samd config` command reports errors in the `mcf` file, correct them. Then repeat the preceding step.
17. Remount the affected file systems.
18. Monitor file system operations.

Restoring the Configuration Without Backup Information

If you lack backup files and/or SAMreports, reconstruct the configuration using whatever information is available. Then proceed as for a new configuration. See the *StorageTek Storage Archive Manager and StorageTek QFS Installation and Configuration Guide* for instructions.

Recovering File Systems

This section outlines the recovery processes that you use when an entire SAM-QFS file system is corrupted or lost. The procedures vary, depending on the type of file system involved and the type of backup and recovery preparations that you have made. But there are two basic tasks that you have to perform:

- [Recreating the File System](#)
- [Restoring Directories and Files](#).

Before you begin, please note: if you are recovering from the loss of a SAM-QFS metadata server, make sure that you have finished [Restoring the SAM-QFS Configuration](#), as described in [Chapter 3](#), before proceeding further. The procedures in this chapter assume that the SAM-QFS software is installed and configured as it was prior to the loss of the file system.

Recreating the File System

Before you can recover files and directories, you must have somewhere to put them. So the first step in the recovery process is to create an empty, replacement file system. Proceed as follows:

Recreate the File System Using Backup Configuration Files and the `sammkfs` Command

1. Log in to the file-system metadata server as root.

```
root@solaris:~#
```
2. Unmount the file system, if it is currently mounted. Use the command `umount mount-point`, where *mount-point* is the directory on which the file system is mounted.
3. Open the `/etc/opt/SUNWsamfs/mcf` file in a text editor, and check the hardware configuration. If you have had to change hardware, edit the file and save the changes.

In the example, we replace the equipment identifiers for two failed disk devices with those of their replacements. Note that the equipment ordinals remain unchanged:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
```

# Equipment	Equipment	Equipment	Family	Device	Additional
# Identifier	Ordinal	Type	Set	State	Parameters
#-----	-----	-----	-----	-----	-----
samqfs1	100	ms	samqfs1	on	

```

/dev/dsk/c1t3d0s3      101      md      samqfs1      on
/dev/dsk/c1t4d0s5      102      md      samqfs1      on
# Tape library
/dev/scsi/changer/c1t2d0 800      rb      lib800      on      ../lib800_cat
/dev/rmt/0cbn          801      li      lib800      on
/dev/rmt/1cbn          802      li      lib800      on
:wq
root@solaris:~#

```

4. Check the mcf file for errors. Use the command `sam-fsd`.

The `sam-fsd` command reads SAM-QFS configuration files and initializes the software. It will stop if it encounters an error:

```
root@solaris:~# sam-fsd
```

5. If the `sam-fsd` command finds an error in the mcf file, edit the file to correct the error and recheck as described in the preceding step.

In the example below, `sam-fsd` reports an unspecified problem with a device. This is probably a typo in an equipment identifier field:

```

root@solaris:~# sam-fsd
Problem in mcf file /etc/opt/SUNWsamfs/mcf for filesystem samqfs1
sam-fsd: Problem with file system devices.
root@solaris:~#

```

6. When the `sam-fsd` command runs without error, the mcf file is correct. Proceed to the next step.

The example is a partial listing of error-free output:

```

root@solaris:~# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal ipc misc proc date
              size    10M age 0
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
root@solaris:~#

```

7. Tell the SAM-QFS software to read the mcf file and reconfigure itself accordingly:

```
root@solaris:~# samd config
```

8. Create the replacement file system. Use the command `sammkfs family-set-name`, where *family-set-name* is the name of the file system.

In the example, we recreate file system `samqfs1`:

```
root@solaris:~# sammkfs samqfs1
```

9. Recreate the mount point directory for the file system, if necessary.

In the example, we recreate the directory `/samqfs1`:

```
root@solaris:~# mkdir /samqfs1
```

10. Back up the operating system's `/etc/vfstab` file.

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

11. Open the `/etc/vfstab` file in a text editor. If the `/etc/vfstab` file does not contain mount parameters for the file system that you are restoring, you will have to restore the mount parameters.

In the example, the SAM-QFS server is installed on a replacement host. So the file contains no mount parameters for the file system that we are restoring, `samqfs1`:

```
root@solaris:~# vi /etc/vfstab
#File
#Device    Device    Mount      System    fsck    Mount      Mount
#to Mount  to fsck   Point      Type      Pass    at Boot    Options
#-----
/devices   -         /devices   devfs     -       no         -
/proc      -         /proc      proc      -       no         -
...
```

12. If possible, when you must restore mount parameters, open a backup copy of the original `/etc/vfstab` file and copy the required line into the current `/etc/vfstab` file. When the changes are complete, save the file and close the editor.

In the example, we have a backup copy, `/zfs1/sam_config/20140127/etc/vfstab`. So we copy the line for the `samqfs1` file system from the backup copy and paste it into the current `/etc/vfstab` file:

```
root@solaris:~# vi /zfs1/sam_config/20140127/etc/vfstab.20140127
#File
#Device    Device    Mount      System    fsck    Mount      Mount
#to Mount  to fsck   Point      Type      Pass    at Boot    Options
#-----
/devices   -         /devices   devfs     -       no         -
/proc      -         /proc      proc      -       no         -
...
samqfs1    -         /samqfs1   samfs     -       yes        stripe=1,bg
:q
root@solaris:~# vi /etc/vfstab
#File
#Device    Device    Mount      System    fsck    Mount      Mount
#to Mount  to fsck   Point      Type      Pass    at Boot    Options
#-----
/devices   -         /devices   devfs     -       no         -
/proc      -         /proc      proc      -       no         -
...
samqfs1    -         /samqfs1   samfs     -       yes        stripe=1,bg
:wq
root@solaris:~#
```

13. Mount the file system.

In the example, we mount the file system `samqfs1`:

```
root@solaris:~# mount /samqfs1
```

14. Now, start [Restoring Directories and Files](#).

Restoring Directories and Files

Once you have recreated the base file system, you can start to restore directories and files. There are two possible approaches:

- [Restoring Files and Directories from a samfsdump \(qfsdump\) Recovery Point File](#) is by far the best option, if you have created and safely stored recovery points on a regular basis.

This approach returns the file system to full functionality immediately, because it restores the file-system metadata. An archiving file system can immediately access data on archival media and stage files back to the disk cache, either immediately or as-needed, when users access files. Files are restored with their original attributes.

If the recovery point contains data as well as metadata, this approach is also the only way to restore stand-alone (non-archiving) file systems that are not backed up by third-party applications.

- [Restoring Files and Directories from Archival Media without a Recovery Point File](#) using a recovery script and the SAM-QFS `star` utility.

Restoring Files and Directories from a `samfsdump` (`qfsdump`) Recovery Point File

Whenever possible, you should base file-system recovery efforts on the most recent available recovery point file. This approach is by far the fastest, most reliable, most thorough, and least labor-intensive way of recovering from the failure of a SAM-QFS file system. So, if a recovery point file exists, proceed as follows:

Restore the Lost File System from a Recovery Point File

1. Log in to the file-system metadata server as `root`.

```
root@solaris:~#
```
2. If you have not already done so, stop archiving and recycling using the procedures in ["Stopping Archiving and Recycling Processes"](#) on page 2-1.
3. Identify the most recent available recovery point file.

In the example, we have been creating dated recovery point files for the file system `samqfs1` in a well-known location, the subdirectory `samqfs1_recovery` on the independent file system `/zfs1`. So the latest file is easy to find:

```
root@solaris:~# mkdir /zfs1/samqfs1_recovery/
20140321 20140322 20140323 20140324
root@solaris:~#
```

4. Change to the mount-point directory for the recreated file system.

```
root@solaris:~# cd /samqfs1
root@solaris:~#
```
5. Restore the entire file system relative to the current directory. Use the command `samfsrestore -T -f recovery-point-file -g logfile` or the QFS-only command `qfsrestore -T -f recovery-point-file -g logfile`, where:
 - `-T` displays recovery statistics when the command terminates, including the number of files and directories processed and the number of errors and warnings.
 - `-f recovery-point-file` specifies the path and file name of the selected recovery point file.
 - `-g logfile` creates a list of the directories and files that were online when the recovery point was created and saves the list to the file specified by `logfile`. If you are restoring an archiving file system, this file can be used to automatically stage files from archival media, so that the disk cache is in the same state as it was at the time that the recovery point was created.

In the example, we restore the file system `samqfs1` from the recovery point file `/zfs1/samqfs1_recovery/20140324` and log the online files in the file `/root/20140324.log`:

```
root@solaris:~# samfsrestore -T -f /zfs1/samqfs1_recovery/20140324 \
-g /root/20140324.log
    samfsdump statistics:
        Files:                52020
        Directories:          36031
        Symbolic links:       0
        Resource files:        8
        File segments:         0
        File archives:         0
        Damaged files:         0
        Files with data:       24102
        File warnings:         0
        Errors:                0
        Unprocessed dirs:      0
        File data bytes:       0
root@solaris:~#
```

6. If you have restored a standalone (non-archiving) file system, the file-system metadata and file data that were saved in the recovery-point file have been restored. Stop here.
7. Next, [Restage Archived Files If Required](#).

Restage Archived Files If Required

1. In most cases, do not restage files from archival media to disk following a file system recovery. Let users stage files as needed, by accessing them.

This approach automatically prioritizes staging according to user needs and maximizes the availability of the file system at a time when it may have been offline for some time. Only required files are staged, and the total staging effort is usually spread over a period of time. This helps to insure that file system resources, such as drives, are always available for high priority tasks, such as archiving new files and staging urgently required user data. This approach also reduces the administrative effort associated with recovery.

2. If you must restage the files that were resident in the disk cache prior to a failure, use the command `/opt/SUNWsamfs/examples/restore.sh logfile`, where *logfile* is the path and file name of the log file that you created with the `-g` option of the `samfsrestore` (`qfsrestore`) command.

The `restore.sh` script stages the files listed in the log file. These are the files that were online when the `samfsrestore` (`qfsrestore`) recovery point file was created.

If thousands of files need to be staged, consider splitting the log file into smaller files and running the `restore.sh` script with each in turn. This spreads the staging effort over a period of time and reduces interference with archiving and user-initiated staging.

3. Now [Identify Damaged Files and Locate Replacement Copies](#).

Identify Damaged Files and Locate Replacement Copies

If you have to restore an archiving file system from a recovery point file that is not completely current, the resulting file system may contain *damaged* files. In a damaged file, the metadata does not point to locations in the archive that hold copies of valid file

data. The restored metadata may point to media that have been recycled, damaged, lost, or replaced since the recovery point was created.

Damaged files are irretrievably lost if there are indeed no archived copies of the data. But, if files were archived *after* a given recovery point was created, the archive may contain copies that the recovery point metadata does not record. So you may be able to undamage and recover at least some of your damaged files. Proceed as follows:

1. If you have not already done so, log in to the file-system metadata server as root.

```
root@solaris:~#
```

2. Identify the most recent available archiver log file.

If the archiver log on the server is still available, it is likely to contain the most recent information. Otherwise, you will need to use a backup copy.

In the example, we have the archive log for the file system, `samqfs1.archiver.log`, on the server in `/var/adm/` and we have been maintaining dated archiver log file copies in a well-known location, the subdirectory `samqfs1_recovery/archlogs` on the independent file system `/zfs1`. So the we have both the latest file, `samqfs1.archiver.log`, and a recent backup, `20140324`:

```
root@solaris:~# dir /zfs1/samqfs1_recovery/archivlogs
20140322    20140323    20140324
root@solaris:~#
```

3. In the newly restored file system, identify any damaged files. Use the command `sfind mountpoint -damaged`, where *mountpoint* is the directory where the recovered file system is mounted.

In the example, we start the search in the directory `/samqfs1` and find six damaged files:

```
root@solaris:~# sfind /samqfs1 -damaged
./genfiles/ay0
./genfiles/ay1
./genfiles/ay2
./genfiles/ay5
./genfiles/ay6
./genfiles/ay9
root@solaris:~#
```

4. Search the most recent copy of the archiver log for entries relating to each of the damaged files. Use the command `grep "file-name-expression" archiver-log`, where *file-name-expression* is a regular expression that matches the damaged file and *archiver-log* is the path and name of the archiver log copy that you are examining.

In the example, we search the file `genfiles/ay0`:

```
root@solaris:~# grep "genfiles\ay0 " /var/adm/samqfs1.archiver.log
```

5. When you find an entry for a file, note the media type, volume serial number, and position of the archive (tar) file where the data file is archived. Also note the file type, since this will affect how you restore the file.

In the example, we locate the file `genfiles/ay0`. The log entry shows that it was archived (A) on March 4, 2014 at 9:49 PM using LTO (1i) volume `VOL012`. The file is stored in the archive file located at position `78 (0x78)`. It is a regular file, type `f`:

```
root@solaris:~# grep "genfiles\ay0 " /var/adm/samqfs1.archiver.log
A 2014/03/04 21:49:15 1i VOL012 SLOT12 allsets.1 78.1 samqfs1 7131.14 8087
```

```
genfiles/ay0 f 0 51
root@solaris:~#
```

For a full explanation of the fields in archiver log entries, see [Appendix A, "Understanding the Archiver Log"](#).

6. If you do not find an entry for a damaged file in the current archiver log copy, repeat the search using any backup archive logs that were created after the recovery point file was created.

Archiver logs are rolled over frequently. So, if you retain multiple archiver log copies, you may be able to recover damaged files using archive copies that were made before the period covered by the current archiver log.

7. Next, [Identify Files that Were Archived After the Recovery Point Was Created](#).

Identify Files that Were Archived After the Recovery Point Was Created

Typically, some files are archived after the last recovery point was created and prior to the loss of a file system. Since the metadata for these files are not in the recovery point file, the files are not recovered when you run `samfsrestore`. But the files do reside on archival media, so you can recover them to their proper place in the file system using the archive logs.

1. If you have not already done so, log in to the file-system metadata server as `root`.

```
root@solaris:~#
```

2. Identify the most recent available archiver log file.

If the archiver log on the server is still available, it is likely to contain the most recent information. Otherwise, you will need to use a backup copy.

In the example, we have the archive log for the file system, `samqfs1.archiver.log`, on the server in `/var/adm/` and we have been maintaining dated archiver log file copies in a well-known location, the subdirectory `samqfs1_recovery/archlogs` on the independent file system `/zfs1`. So the we have both the latest file, `samqfs1.archiver.log`, and a recent backup, `20140324`:

```
root@solaris:~# dir /zfs1/samqfs1_recovery/archivlogs
20140322    20140323    20140324
root@solaris:~#
```

3. Search the most recent copy of the archiver log for entries that were made after the recovery point was created. Use the command `grep "time-date-expression" archiver-log`, where *time-date-expression* is a regular expression that matches the date and time where you want to start searching and *archiver-log* is the path and name of the archiver log copy that you are examining.

In the example, we lost the file system at 2:02 AM on March 4, 2014. The last recovery point file was made at 2:10 AM on March 3, 2014. So we search for archived files that were logged on March 3 or 4.

```
root@solaris:~# grep "^A 2014\|03\|[34]" /var/adm/samqfs1.archiver.log
```

4. When you find an entry for a backup copy of an unrestored file, note the path, name, file type, media type, and location information.

File types are listed as `f` for regular files, `R` for removable-media files, or `S` for a data segment in a segmented file. The media type is a two-character code (see [Appendix B](#)).

To locate the backup copy, you need the volume serial number of the media volume that stores the copy. If the copy is stored on sequential-access media, such as magnetic tape, also note the hexadecimal value that represents the starting position of the archive (tar) file. If the copy is stored on random-access media, such as archival disk, note the path and file name of the tar file relative to the volume serial number. Finally, if the file is segmented, note the segment length.

In the example below, the archiver log entries show that the following files were archived after the last recovery point was created:

```
root@solaris:~# grep "^A 2014\03\[34]" /var/adm/samqfs1.archiver.log
A 2014/03/03 10:43:18 li VOL002 all.1 111.1 samqfs1 1053.3 69 genfiles/hops f 0
0
A 2014/03/03 10:43:18 li VOL002 all.1 111.3 samqfs1 1051.1 104 genfiles/anic f
0 0
A 2014/03/03 13:09:05 li VOL004 all.1 212.1 samqfs1 1535.2 1971 genfiles/genA0
f 0 0
A 2014/03/03 13:09:06 li VOL004 all.1 212.20 samqfs1 1534.2 1497 genfiles/genA9
f 0 0
A 2014/03/03 13:10:15 li VOL004 all.1 212.3f samqfs1 1533.2 6491 genfiles/genA2
f 0 0
A 2014/03/03 13:12:25 li VOL003 all.1 2.5e samqfs1 1532.2 17717 genfiles/genA13
f 0 0
A 2014/03/03 13:12:28 li VOL003 all.1 2.7d samqfs1 1531.2 14472 genfiles/genA4
f 0 0
A 2014/03/03 13:12:40 li VOL003 all.1 2.9c samqfs1 1530.2 19971 genfiles/genA45
f 0 0
A 2014/03/03 21:49:15 dk DISKVOL1/f2 all.1 2.2e9 samqfs1 1511.2 8971
socfiles/spcC4 f 0 0
A 2014/03/03 21:49:15 dk DISKVOL1/f2 all.1 2.308 samqfs1 1510.2 7797
spcfiles/spcC5 f 0 0
A 2014/03/03 14:01:47 li VOL013 all.1 76a.1 samqfs1 14.5 10485760 bf/dat011/1 S
0 51
A 2014/03/03 14:04:11 li VOL013 all.1 76a.5002 samqfs1 15.5 10485760
bf/dat011/2 S 0 51
A 2014/03/03 14:06:24 li VOL013 all.1 1409aa4.1 samqfs1 16.5 184 bf/dat011/3 S
0 51
A 2014/03/03 18:28:51 li VOL036 all.1 12d.1 samqfs1 11731.1 89128448 rf/rf81 f
0 210
A 2013/08/23 18:28:51 li VOL034 all.1 15f.0 samqfs1 11731.1 525271552 rf/rf81 f
1 220
root@solaris:~#
```

We note the following information:

- Eight regular (type f) files are archived (A) on LTO (li) media: genfiles/hops and genfiles/anic at position 0x111 on volume VOL002, genfiles/genA0, genfiles/genA9 and genfiles/genA2 at position 0x212 on volume VOL004, and genfiles/genA13, genfiles/genA4, and genfiles/genA45 at position 0x212 on volume VOL003.
- Two regular (type f) files are archived (A) on disk (dk) media: spcfiles/spcC4 and spcfiles/spcC5 in archive file DISKVOL1 \f2 on volume DISKVOL1.
- One, three-part, segmented (type S) file is archived on LTO (li) media: bf/dat011, in two segments starting at position 0x76a and one segment starting at position 1409aa4 on volume VOL013. Segment /1 is 10485760 bytes long, segment /2 is 10485622 bytes, and segment /3 is 184 bytes.

- One, regular (type `f`), volume overflow file archived (A) on LTO (11) media: `rf/rf81`, starting at position `0x12d` on volume `VOL036` and continuing at position `0x15f` on volume `VOL034`.

For a full explanation of the fields in archiver log entries, see [Appendix A, "Understanding the Archiver Log"](#).

5. Repeat the search using any backup archive logs that were created after the recovery point file was created.

Archiver logs are rolled over frequently. So, if you retain multiple archiver log copies, you may be able to recover damaged files using archive copies that were made before the period covered by the current archiver log.

6. Now [Restore the Damaged and/or Missing Files](#).

Restore the Damaged and/or Missing Files

Given the media volume and the position of an archive (`tar`) file on the media, restoring a missing or damaged files is simply a matter of accessing the `tar` file and extracting the required data file. When the archive files reside on archival disk devices, this is simple, because the `tar` files reside in randomly accessible directories under a file-system mount point. When the `tar` file resides on high-capacity, sequential-access media like tape, however, there is an added complication: we cannot normally extract the required data file from the archive file until the latter is staged to a random-access disk device. Since archive files can be large, this can be time-consuming and awkward in a recovery situation. So the procedures below take advantage of the SAM-QFS command `request`, which reads the archive files into memory and makes them available as if they were being read from disk.

Restore as many damaged and missing regular files as you can. For each file, proceed as follows:

1. Start by recovering regular files that do not span volumes. Use the procedure "[Restore Lost and Damaged Regular Files](#)" on page 5-3.
2. Next, recover the segmented files. Use the procedure "[Restore Lost and Damaged Segmented Files](#)" on page 5-4.
3. Then restore the regular files that do span volumes. Use the procedure "[Restore Lost and Damaged Volume Overflow Files](#)" on page 5-7.
4. Once you have restored all missing and damaged files that have copies, re-enable archiving by removing `wait` directives from the `archiver.cmd` file and re-enable recycling by removing `-ignore` parameters from the `recycler.cmd` file.

The file system is as close to its original condition as possible. Files that are still damaged or missing cannot be recovered.

5. Once you have restored all missing and damaged files that have copies, go to "[Restoring Archiving File Systems to Normal Operation](#)" on page 6-1.

Restoring Files and Directories from Archival Media without a Recovery Point File

If you must recover a file system directly from the archival media, without the assistance of a recovery point file, you can do so. You proceed as follows:

1. If you are trying to restore files from optical media, stop here and contact Oracle support services for assistance.
2. Disable network file system (NFS) sharing for the file system.

3. Disable archiving and recycling, using the method outlined in ["Stopping Archiving and Recycling Processes"](#) on page 2-1.
4. Reserve a tape drive for the exclusive use of the recovery process. Use the command `samcmd unavail drive-equipment-number`, where *drive-equipment-number* is the equipment ordinal number assigned to the drive in the `/etc/opt/SUNWsamfs/mcf` file.

The `samcmd unavail` command makes the drive unavailable to archiving, staging and releasing processes. In the example, we reserve drive 804

```
root@solaris:~# samcmd unavail 804
root@solaris:~#
```

5. Copy the file `/opt/SUNWsamfs/examples/tarback.sh` to an alternate location, such as `/tmp`.

The `tarback.sh` file is an executable script that restores files from a specified set of media volumes. The script runs the command `star -n` against each archive (tar) file on each volume. When a backup copy on tape has no corresponding file in the file system or when the copy on tape is newer than the corresponding file in the file system, `star -n` restores the copy.

In the example, we copy the script to `/tmp`:

```
root@solaris:~# cp /opt/SUNWsamfs/examples/tarback.sh /tmp/tarback.sh
root@solaris:~#
```

6. Open the copy of the `tarback.sh` file in a text editor.

```
root@solaris:~# vi /opt/SUNWsamfs/examples/tarback.sh
#!/bin/sh
# script to reload files from SAMFS archive tapes
STAR="/opt/SUNWsamfs/sbin/star"
LOAD="/opt/SUNWsamfs/sbin/load"
UNLOAD="/opt/SUNWsamfs/sbin/unload"
EQ=28
TAPEDRIVE="/dev/rmt/3cbn"
# BLOCKSIZE is in units of 512 bytes (e.g. 256 for 128K)
BLOCKSIZE=256
MEDIATYPE="lt"
VSN_LIST="VSNA VSNB VSNC VSNZ"
...
```

7. If the SAM-QFS utilities `star`, `load`, and `unload` are installed in non-standard locations, edit the default command paths in the copy of the `tarback.sh` file.

In the example, all utilities are installed in the default locations, so no edits are needed:

```
root@solaris:~# vi /opt/SUNWsamfs/examples/tarback.sh
#!/bin/sh
# script to reload files from SAMFS archive tapes
STAR="/opt/SUNWsamfs/sbin/star"
LOAD="/opt/SUNWsamfs/sbin/load"
UNLOAD="/opt/SUNWsamfs/sbin/unload"
...
```

8. In the copy of the `tarback.sh` file, locate the variable `EQ` and set its value to the equipment ordinal number of the drive that you reserved for recovery use.

In the example, we set `EQ=804`:

```

root@solaris:~# vi /opt/SUNWsamfs/examples/tarback.sh
#!/bin/sh
#  script to reload files from SAMFS archive tapes
STAR="/opt/SUNWsamfs/sbin/star"
LOAD="/opt/SUNWsamfs/sbin/load"
UNLOAD="/opt/SUNWsamfs/sbin/unload"
EQ=804
...

```

9. In the copy of the `tarback.sh` file, locate the variable `TAPEDRIVE` and set its value to the raw path to the device, enclosed in double quotation marks.

In the example, the raw path to device 804 is `/dev/rmt/3cbn`:

```

root@solaris:~# vi /opt/SUNWsamfs/examples/tarback.sh
#!/bin/sh
#  script to reload files from SAMFS archive tapes
STAR="/opt/SUNWsamfs/sbin/star"
LOAD="/opt/SUNWsamfs/sbin/load"
UNLOAD="/opt/SUNWsamfs/sbin/unload"
EQ=804
TAPEDRIVE="/dev/rmt/3cbn"
...

```

10. In the copy of the `tarback.sh` file, locate the variable `BLOCKSIZE` and set its value to the number of 512-byte units in the desired block size.

In the example, we want a 256-kilobyte segment size for the LTO-4 drive. So we specify 512:

```

LOAD="/opt/SUNWsamfs/sbin/load"
UNLOAD="/opt/SUNWsamfs/sbin/unload"
EQ=804
TAPEDRIVE="/dev/rmt/3cbn"
BLOCKSIZE=512
...

```

11. In the copy of the `tarback.sh` file, locate the variable `MEDIATYPE` and set its value to the two-character media-type code that [Appendix B](#) lists for the type of media that the drive supports. Enclose the media type in double quotation marks.

In the example, we are using an LTO-4 drive. So we specify `lt`:

```

EQ=804
TAPEDRIVE="/dev/rmt/3cbn"
BLOCKSIZE=512
MEDIATYPE="lt"
...

```

12. In the copy of the `tarback.sh` file, locate the variable `VSN_LIST` and, as its value, supply a space-delimited list of the volume serial numbers (VSNs) of the tape volumes that might contain backup copies of your files, enclosed in double quotation marks.

In the example, we specify volumes `VOL002`, `VOL003`, `VOL004`, `VOL013`, `VOL034`, and `VOL036`:

```

EQ=804
TAPEDRIVE="/dev/rmt/3cbn"
BLOCKSIZE=512
MEDIATYPE="lt"
VSN_LIST="VOL002 VOL003 VOL004 VOL013 VOL034 VOL036"

```

...

13. Save the copy of the `tarback.sh` file and close the editor.

```
EQ=804
TAPEDRIVE="/dev/rmt/3cbn"
BLOCKSIZE=512
MEDIATYPE="lt"
VSN_LIST="VOL002 VOL003 VOL004 VOL013 VOL034 VOL036"
...
:wq
root@solaris:~#
```

14. Execute the `/tmp/tarback.sh` script.

```
root@solaris:~# /tmp/tarback.sh
```

15. For each restored file, recreate user and group ownership, modes, extended attributes, and access control lists (ACLs) as necessary.

The `/tmp/tarback.sh` script cannot restore these types of metadata.

16. Once you have run the `/tmp/tarback.sh` script and finished recovering files, go to ["Restoring Archiving File Systems to Normal Operation"](#) on page 6-1.

Recovering Lost and Damaged Files

This chapter outlines procedures for restoring individual files to the file system. It covers the following topics:

- [Recovering Files Using a Recovery Point File](#)
- [Recovering Files Using Archiver Log Entries](#)
- [Recovering Damaged Archive Copies](#)

Recovering Files Using a Recovery Point File

A recovery point file is the fastest, most reliable, most thorough, and least labor-intensive way of recovering lost or damaged files. So, if a recovery point file is available, proceed as follows:

1. Log in to the file-system metadata server as root.

```
root@solaris:~#
```

2. If you have not already done so, stop archiving and recycling using the procedure in "[Stopping Archiving and Recycling Processes](#)" on page 2-1
3. In the target file system, create a temporary recovery directory to hold the recovered files.

In the example, we create the temporary directory `restore` under the mount point for the recreated file system, `/samqfs1`:

```
root@solaris:~# mkdir /samqfs1/restore
```

4. Keep the archiver from archiving from the temporary directory. Use the command `archive -r -n directory`, where *directory* is the path and directory name of the temporary directory.

```
root@solaris:~# archive -r -n /samqfs1/restore
```

5. Change to the temporary recovery directory.

```
root@solaris:~# cd /samqfs1/restore
```

6. Identify the most recent available recovery point file.

In the example, we have been creating dated recovery point files for the file system `samqfs1` in a well-known location, the subdirectory `samqfs1_recovery` on the independent file system `/zfs1`. So the latest file is easy to find:

```
root@solaris:~# dir /zfs1/samqfs1_recovery/
20140321    20140322    20140323    20140324
```

```
root@solaris:~#
```

7. Make sure that the file that you need to recover is in the recovery point file. Use the command `samfsrestore -t -f recovery-point`, where `-t` specifies the table of contents and `recovery-point-file` specifies the path and file name of the selected recovery point file. Search the output for the required file.

In the example, we are trying to recover the file `genw445`. So we specify the recovery point file `/zfs1/samqfs1_recovery/20140324` and pipe the output of `samfsrestore -t` to the Solaris `grep` command and the regular expression `"genw445"`:

```
root@solaris:~# samfsrestore -t -f /zfs1/samqfs1_recovery/20140324 | \
grep "genw445"
./genfiles/genw445
root@solaris:~#
```

8. Restore the file's inode information to the current directory. Use the command `samfsrestore -f recovery-point file`, where `file` specifies the exact path and name that the recovery point file lists for the file that you want to recover.

In the example, we recover `./genfiles/genw445`:

```
root@solaris:~# samfsrestore -f /zfs1/samqfs1_recovery/20140324 \
./genfiles/genw445
root@solaris:~#
```

9. Make sure that the file has been restored correctly. Use the command `sls -D file`, where `file` specifies the path and name of the file relative to the temporary recovery directory.

In the example, the file `genfiles/genw445` has been recovered:

```
root@solaris:~# sls -D genfiles/genw445
genfiles/genw445:
  mode: -rw-r--r--   links:   1  owner: data           group: samqfs1
  length:      14975  inode:   25739.1
offline; archdone;
copy 1: ---- Mar  4 11:55 8ae.1 xt 000000
copy 2: ---- Mar  4 15:51 cd3.7f57 xt 000000
  access:      Mar  4 11:55  modification: Mar  4 21:50
  changed:      Mar  4 11:50  attributes:   Mar  4 21:50
  creation:     Mar  4 11:50  residence:    Mar  4 21:50
root@solaris:~#
```

10. Move the recovered file to the correct location in the file system.

```
root@solaris:~# mv -f genfiles/genw445 /samqfs1/genfiles/genw445
root@solaris:~#
```

11. Repeat this procedure until all missing files have been recovered.
12. Finish the recovery procedure. Go to ["Restoring Archiving File Systems to Normal Operation"](#) on page 6-1.

Recovering Files Using Archiver Log Entries

Using an archive log for guidance when recovering files from media is always a somewhat tedious and labor-intensive process, if more than a few files are involved. So, whenever possible, use the procedures in this section only when a recovery point cannot restore the file that you need.

While the process of recovering files from archival media is essentially the same in all cases, some details differ significantly, depending on the type of file that you are recovering. So select the procedure intended for the type of file that you are restoring:

- [Restore Lost and Damaged Regular Files](#)
- [Restore Lost and Damaged Segmented Files](#)
- [Restore Lost and Damaged Volume Overflow Files.](#)

Note that files may not be restored to the precise location that you expect when you recover a copy from the media. Files are restored to their location at the time when the archive copy was made. So files that were subsequently moved are not restored to the directory where they were when they were lost.

Restore Lost and Damaged Regular Files

For each file that you need to recover, proceed as follows:

1. If you have not already done so, log in to the file-system metadata server as root.

```
root@solaris:~#
```

2. If you have not already done so, stop archiving and recycling using the procedure in ["Stopping Archiving and Recycling Processes"](#) on page 2-1
3. Change to the root directory of the file system that you are restoring.

SAM-QFS archive files store copies relative to the file-system root directory. So to restore them to their original locations, we want to restore them from the root directory.

In the example, we change to the root of the `samqfs1` file system.

```
root@solaris:~# cd /samqfs1
```

4. If you have an archiver log for the period when the regular file was last archived, find the most recent entry for the file. Note the media type, the volume serial number of the media, and the path and name of the file relative to the root directory of the file system. If the copy is stored on sequential-access media, such as magnetic tape, also note the hexadecimal value that represents the starting position of the archive (tar) file. If the copy is stored on random-access media, such as archival disk, note the path and file name of the tar file relative to the volume serial number.

```
A 2014/03/03 13:09:05 li VOL004 all.1 212.1 samqfs1 1535.2 1971 genfiles/genA0
f 0 0
A 2014/03/03 21:49:15 dk DISKVOL1/f2 all.1 2.2e9 samqfs1 1511.2 8971
socfiles/spcC4 f 0 0
```

5. If the file that you are restoring is archived on disk media, extract the archive copy of the missing or damaged file from the tar file on the disk volume. Use the command `star -xv -f tarfile file`, where `tarfile` is the name of the archive file and `file` is the path—relative to the file-system root directory—and name of the file that you need to restore. Stop here, and go on to the next file.

The `star` command is an enhanced SAM-QFS version of GNU `tar` that restores specified files from the archive file that you are pointing to with the request file.

In the example, we extract the data file `genfiles/genC4` from the tar file `DISKVOL1/f2`. The file is restored to `/samqfs1/genfiles/genC4`:

```
root@solaris:~# star -xvf DISKVOL1/f2 genfiles/genC4
```

6. If the file that you are restoring is archived on removable media, such as magnetic tape, create a directory in the restored file system to hold temporary archive files.

In the example, we create the directory `/samqfs1/tars`

```
root@solaris:~# mkdir /samqfs1/tars
```

7. Position the media at the beginning of the tar header for the archive file that holds the archived copy, and read the archive from the media into memory. Use the command `request -m media-type -v volume-serial-number -p 0xposition path/requestfile`, where:

- *media-type* is one of the two-character media type codes listed in [Appendix B](#)
- *volume-serial-number* is the six-character alphanumeric code that identifies the media volume
- *position* is the hexadecimal starting position that you noted in the archiver log entry
- *path* is the path to the temporary recovery directory.
- *requestfile* is the name that you want to use for the in-memory tar file that the request command reads from the media.

In the example, we create a request file, `/samqfs1/tars/currentrequest` starting from position `0x78` on LTO (1i) VOL012:

```
root@solaris:~# request -m li -v VOL012 -p 0x78 /samqfs1/tars/currentrequest
```

8. Extract the archive copy of the missing or damaged file from the in-memory tar file that you created in the preceding step. Use the command `star -xv -f requestfile`, where *requestfile* is the name of the in-memory tar file and *file* is the path—relative to the file-system root directory—and name of the file that you need to restore.

The star command is an enhanced SAM-QFS version of GNU tar that restores specified files from the archive file that you are pointing to with the request file.

In the example, we extract the data file `genfiles/genA0` from the request file (in-memory tar file) `currentrequest` which is in the temporary directory `tars/`. The file is restored to `/samqfs1/genfiles/genA0`:

```
root@solaris:~# star -xvf tars/currentrequest genfiles/genA0
```

9. Set any required file attributes.

When you restore a file from a tar file, without a `samfsdump` or `qfsdump` recovery point file, the original file attributes are lost. An `.inodes` file has to be created for the file from scratch, using default attribute values.

10. Repeat this procedure until all required files have been recovered.
11. Finish the recovery procedure. Go to ["Restoring Archiving File Systems to Normal Operation"](#) on page 6-1.

Restore Lost and Damaged Segmented Files

Restoring a segmented file is much like restoring a regular file. However, you recover the individual segments rather than the file itself. So, to restore the file, you must reassemble the segments into a single file and then re-segment the result. For each file that you need to recover, proceed as follows:

1. If you have not already done so, log in to the file-system metadata server as root.

```
root@solaris:~#
```

2. If you have not already done so, stop archiving and recycling using the procedure in ["Stopping Archiving and Recycling Processes"](#) on page 2-1
3. If you have an archiver log for the period when the segmented file was last archived, find the most recent entry for the file. Note the media type, the volume serial number(s) of the media, the hexadecimal position where the archive (tar) file starts, file, the path and name of the segmented file relative to the root directory of the file system, and the number of segments in the file.

```
A 2014/03/03 14:01:47 li VOL013 all.1 76a.1 samqfs1 14.5 10485760 bf/dat011/1 S
0 51
A 2014/03/03 14:04:11 li VOL013 all.1 76a.5002 samqfs1 15.5 10485760
bf/dat011/2 S 0 51
A 2014/03/03 14:06:24 li VOL013 all.1 1409AA4.1 samqfs1 16.5 184 bf/dat011/3 S
0 51
```

4. Change to the root directory of the file system that you are restoring.

SAM-QFS archive files store copies relative to the file-system root directory. So to restore them to their original locations, we want to restore them from the root directory.

In the example, we change to the root of the samqfs1 file system.

```
root@solaris:~# cd /samqfs1
```

5. Create a directory in the restored file system to hold temporary archive files.

In the example, we create the directory /samqfs1/tars

```
root@solaris:~# mkdir /samqfs1/tars
```

6. Position the media at beginning of each archive file that holds an archived copy of one or more of the file segments, and read the archive from the media into memory. Use the command `request -m media-type -v volume-serial-number -p 0xposition path/requestfile`, where:

- *media-type* is one of the two-character media type codes listed in [Appendix B](#)
- *volume-serial-number* is the six-character alphanumeric code that identifies the media volume
- *position* is the hexadecimal starting position that you noted in the archiver log entry
- *path* is the path to the temporary recovery directory.
- *requestfile* is the name that you want to use for the in-memory tar file that the request command reads from the media.

In the example, we need to create two request files. The first, /samqfs1/tars/request76a, loads the archive file that starts at position 0x76a on LTO (li) VOL013. This archive contains both of the first two segments. The second request file, /samqfs1/tars/request1409aa4, loads the archive file at position 0x1409aa4, in this case on the same volume (segments can reside on any volume in the library):

```
root@solaris:~# request -m li -v VOL013 -p 0x76a /samqfs1/tars/request76a
root@solaris:~# request -m li -v VOL013 -p 0x1409aa4 \
/samqfs1/tars/request1409aa4
```

7. Extract each segment of the backup copy of the missing or damaged file from the in-memory tar file that you created in the preceding step. Use the command `star -xv -f requestfile segment`, where *requestfile* is the name of the in-memory tar file and *segment* is the path—relative to the file-system root directory—and name of each segment of the file that you need to restore.

The star command is an enhanced SAM-QFS version of GNU tar that restores specified files from the archive file that you are pointing to with the request file.

In the example, we extract two of the three segments of the data file `bf/dat011` from the request file (in-memory tar files) `tars/request76a` and one from the request file `tars/request1409aa4`. The file is restored to a directory, `/samqfs1/bf/dat011/`, in three separate pieces:

```
root@solaris:~# star -xvf tars/request76a bf/dat011/1
root@solaris:~# star -xvf tars/request76a bf/dat011/2
root@solaris:~# star -xvf tars/request1409aa4 bf/dat011/3
When we list the contents of /samqfs1/bf/dat011, we see one sequentially
numbered file for each restored segment:
```

```
root@solaris:~# ls /samqfs/bf/dat011
total 40968
-rw-rw---- 1 root other 10485760 Mar  5 17:06 1
-rw-rw---- 1 root other 10485760 Mar  5 17:06 2
-rw-rw---- 1 root other      184 Mar  5 17:07 3
root@solaris:~#
```

8. Re-assemble the restored segments into a single, unsegmented, temporary file.

In the example, we concatenate the three segments in the `/samqfs1/bf/dat011/` directory to create the file `/samqfs1/bf/dat011file`:

```
root@solaris:~# cat /samqfs/bf/dat011/1 /samqfs/bf/dat011/2 \
/samqfs/bf/dat011/3 > /samqfs/bf/dat011file
root@solaris:~#
```

When we list the contents of `/samqfs1/bf/`, the new file appears alongside the directory containing the segments.

```
root@solaris:~# ls -l /samqfs/bf/dat011*
drwxr-xr-x 2 root root      4096 Mar  5 17:06 dat011
-rw-rw---- 1 root other 20971704 Mar  5 17:14 dat011file
root@solaris:~#
```

9. Remove the segments and the directory that contains them.

```
root@solaris:~# rm -r /samqfs/bf/dat011/
root@solaris:~#
```

10. Create an empty file using the original path and name of the segmented file. Use the command `touch file`, where *file* is the original path and file name.

In the example, we create the empty file `/samqfs/bf/dat011`, the original name of the segmented file that we are restoring:

```
root@solaris:~# touch /samqfs/bf/dat011
root@solaris:~#
```

11. Set the SAM-QFS segment attribute on the newly created, empty file. Use the command `segment -l segment-length file`, where *segment-length* is the segment length that you noted in the archiver log entry and *file* is the original path and name of the segmented file.

In the example, the archiver log shows that the segment length for the file `dat011` is 10485760 (the file ends in the third segment, so the length of the data on the media is less than the segment length):

```
A 2014/03/03 14:01:47 li VOL013 all.1 76a.1 samqfs1 14.5 10485760 bf/dat011/1 S
0 51
A 2014/03/03 14:04:11 li VOL013 all.1 76a.5002 samqfs1 15.5 10485760
bf/dat011/2 S 0 51
A 2014/03/03 14:06:24 li VOL013 all.1 1409aa4.1 samqfs1 16.5 184 bf/dat011/3 S
0 51
```

So we set the segment length for the empty file to 10485760:

```
root@solaris:~# segment -l 10485760 /samqfs/bf/dat011
root@solaris:~#
```

12. Copy the unsegmented temporary file to the empty segmented file.

In the example, we copy `dat011file` to `dat011`:

```
root@solaris:~# cp /samqfs/bf/dat011file /samqfs/bf/dat011
root@solaris:~#
```

When we use the command `sls -2K samqfs/bf/dat011` to list the segments, they are listed as shown. So the file has been restored.

```
root@solaris:~# sls -2K /samqfs/bf/dat011
-rw-rw---- 1 root other      20971704    Mar  5 17:12 samqfs/bf/dat011
----- sI {3,0,0,0}
-rw-rw---- 1 root other      10485760    Mar  5 17:12 samqfs/bf/dat011/1
----- sS
-rw-rw---- 1 root other      10485760    Mar  5 17:12 samqfs/bf/dat011/2
----- sS
-rw-rw---- 1 root other        184      Mar  5 17:12 samqfs/bf/dat011/3
----- sS
```

13. Set any other required file attributes.

When you restore a file from a tar file, without a `samfsdump` or `qfstdump` recovery point file, the original file attributes are lost. An `.inodes` file has to be created for the file from scratch, using default attribute values.

14. The file has now been restored. Delete the unsegmented temporary file.

In the example, we delete `dat011file`:

```
root@solaris:~# rm /samqfs/bf/dat011file
root@solaris:~#
```

15. Repeat this procedure until all required files have been recovered.

16. Finish the recovery procedure. Go to ["Restoring Archiving File Systems to Normal Operation"](#) on page 6-1.

Restore Lost and Damaged Volume Overflow Files

A volume overflow file is a regular file that spans media volumes. Restoring a volume overflow file is thus much like restoring any other regular file. However, you must combine sections of an archive file that resides on multiple volumes into a single archive file on disk before you extract the data file from the archive. So, for each file that you need to recover, proceed as follows:

1. If you have not already done so, log in to the file-system metadata server as root.

```
root@solaris:~#
```

2. If you have not already done so, stop archiving and recycling using the procedure in ["Stopping Archiving and Recycling Processes"](#) on page 2-1
3. If you have an archiver log for the period when the volume overflow file was last archived, find the most recent entry for the file. Note the volume serial number(s) of the media, the length of each section of the file, the path and name of the file relative to the root directory of the file system, and the number of sections in the file.

In the example, we know that the file `/samqfs1/rf/rf81` is a volume overflow because it is a regular, type `f` file that resides on two volumes, `VOL036` and `VOL034`, and has two sections, 0 and 1:

```
A 2014/03/03 18:28:51 li VOL036 all.1 12d.1 samqfs1 11731.1 89128448 rf/rf81 f
0 210
A 2013/08/23 18:28:51 li VOL034 all.1 15f.0 samqfs1 11731.1 525271552 rf/rf81 f
1 220
```

4. Change to the root directory of the file system that you are restoring.

SAM-QFS archive files store copies relative to the file-system root directory. So to restore them to their original locations, we want to restore from the root directory.

In the example, we change to the root of the `samqfs1` file system.

```
root@solaris:~# cd /samqfs1
```

5. Before proceeding, make sure that the file system contains enough free space accommodate a file at least twice the size of the file that you are recovering.

For the file in the example, `rf/rf81`, we will need about 1.2 gigabytes of free space, based on the sizes of the two sections of the file: $2 \times (89128448 + 525271552) = 1228800000$ bytes.

6. Create a directory in the restored file system to hold temporary archive files.

In the example, we create the directory `/samqfs1/tars`

```
root@solaris:~# mkdir /samqfs1/tars
```

7. Position the media at beginning of each archive file that holds an archived copy of one or more of the file segments, and read the archive from the media into memory. Use the command `request -m media-type -v volume-serial-number -p 0xposition path/requestfile`, where:

- `media-type` is one of the two-character media type codes listed in [Appendix B](#)
- `volume-serial-number` is the six-character alphanumeric code that identifies the media volume
- `position` is the hexadecimal starting position that you noted in the archiver log entry
- `path` is the path to the temporary recovery directory.
- `requestfile` is the name that you want to use for the in-memory tar file that the request command reads from the media.

In the example, we create two request files. The first request file, `/samqfs1/tars/requestVOL036`, loads the archive file that starts at position `0x12d`

on LTO (1i) VOL036. The second request file, /samqfs1/tars/requestVOL034, loads the archive file at position 0x15f on LTO (1i) VOL034:

```
root@solaris:~# request -m 1i -v VOL036 -p 0x12d /samqfs1/tars/requestVOL036
root@solaris:~# request -m 1i -v VOL034 -p 0x15f /samqfs1/tars/requestVOL034
```

8. Save each of the in-memory tar files that you created to disk as a section of the archive file. Use the command `dd if= requestfile of=archive_section`, where *requestfile* is the path and name of the in-memory tar file and *archive_section* is the path and name of each section of the archive file.

In the example, we save the request files (in-memory tar files), tars/requestVOL036 and tars/requestVOL034 as tars/archive_part1 and tars/archive_part2:

```
root@solaris:~# dd if=tars/requestVOL036 of=tars/archive_part1
root@solaris:~# dd if=tars/requestVOL034 of=tars/archive_part2
root@solaris:~#
```

9. Re-assemble the sections into a single archive file.

In the example, we concatenate the two sections, tars/archive_part1 and tars/archive_part2, to create a single archive file, /tars/archive_complete:

```
root@solaris:~# cat tars/archive_part1 tars/archive_part2 > \
tars/archive_complete
root@solaris:~#
```

10. Extract the backup copy of the missing or damaged volume overflow file from the archive (tar) file that you created in the preceding step. Use the command `star -xv -f tarfile file`, where *tarfile* is the name of the archive file and *file* is the path—relative to the file-system root directory—and name of the volume overflow file that you need to restore.

The star command is an enhanced SAM-QFS version of GNU tar that restores specified files from the archive file that you are pointing to with the request file.

In the example, we extract the volume overflow file rf/rf81 from the tar file tars/archive_complete:

```
root@solaris:~# star -xvf tars/archive_complete rf/rf81
```

11. Set any other required file attributes.

When you restore a file from a tar file, without a samfsdump or qfsdump recovery point file, the original file attributes are lost. An .inodes file has to be created for the file from scratch, using default attribute values.

12. The volume overflow file has now been restored. Delete the temporary file.

In the example, we delete dat011file:

```
root@solaris:~# rm tars/archive_*
root@solaris:~#
```

13. Repeat this procedure until all required files have been recovered.
14. Finish the recovery procedure. Go to ["Restoring Archiving File Systems to Normal Operation"](#) on page 6-1.

Recovering Damaged Archive Copies

A *damaged* archive copy is a copy of a file that cannot be staged back to the disk cache. Sometimes, the file has merely failed to stage due to an intermittent, hardware-related I/O problem and can be easily resolved. At other times, the damaged copy is corrupt and the data is irrecoverable. Your only option in such cases is to recover the file from an alternate copy.

To identify and manage damaged copies, proceed as follows:

1. Identify files that have damaged archive copies. Use the command `sfind mountpoint -any_copy_d`, where *mountpoint* is the directory where the recovered file system is mounted.

In the example, we start the search in the directory `/samqfs1` and find three files that have damaged copies:

```
root@solaris:~# sfind /samqfs1 -any_copy_d
./genfiles/ab09
./genfiles/ab11
./genfiles/ay12
root@solaris:~#
```

2. For each file that you identified, identify the damaged copies. Use the command `sls -D file`, where *file* is the path and file name that you want to check.

Damaged copies are flagged with a D. In the example, **copy 2** of `/samqfs1/genfiles/ab09` and **copy 1** of `/samqfs1/genfiles/ab11` are damaged:

```
root@solaris:~# sls -D /samqfs1/genfiles/ab09
/samqfs1/genfiles/ab09:
mode: -rw-r----- links: 1 owner: root group: other
length: 306581 admin id: 0 inode: 11748.11
project: system(0)
copy 1: ---- Mar 11 13:52 76f.421bc li VOL011
copy 2: ---D Mar 31 14:02 286.1324f li VOL021
access: Mar 11 13:50 modification: Mar 11 13:50
changed: Mar 11 13:50 attributes: Mar 11 13:50
creation: Mar 11 13:50 residence: Mar 11 13:50
root@solaris:~# sls -D /samqfs1/genfiles/ab11
/samqfs1/genfiles/ab11:
mode: -rw-r----- links: 1 owner: root group: other
length: 380051 admin id: 0 inode: 1460.1
project: system(0)
copy 1: ---D Mar 01 10:21 431.21bc6 li VOL024
access: Mar 01 10:21 modification: Mar 01 10:21
changed: Mar 01 10:21 attributes: Mar 01 10:21
creation: Mar 01 10:21 residence: Mar 01 10:21
root@solaris:~#
```

3. If there is an alternate copy, unarchive the damaged copy. Use the command `unarchive -c CopyNumber file`, where *CopyNumber* is an integer representing the copy number and *file* is the path and file name of the damaged file. Stop here.

When you unarchive the damaged copy, SAM-QFS stages from the remaining copy and creates an additional archive copy the next time that the archiver process runs. In the example, we have another, undamaged copy of `/samqfs1/genfiles/ab09`, so we unarchive the damaged copy, copy 2:

```
root@solaris:~# unarchive -c 2 /samqfs1/genfiles/ab09
root@solaris:~#
```

4. If you do not have another copy, undamage the damaged copy. Use the command `undamage -c CopyNumber file`, where *CopyNumber* is an integer representing the copy number and *file* is the path and file name of the damaged file.

Sometimes a file fails to stage due to an intermittent, hardware-related I/O error. Clearing the damage flag and restaging may solve the problem. In the example, there is only one copy of `/samqfs1/genfiles/ab11`:

```
root@solaris:~# undamage -c1 /samqfs1/genfiles/ab11
```

5. Try to stage the copy. Use the command `stage -c CopyNumber -I file`, where *CopyNumber* is an integer representing the copy number and *file* is the path and file name of the file.

The optional `-I` (immediate) parameter pushes the staging operation to the head of the queue:

```
root@solaris:~# stage -c 1 -I /samqfs1/genfiles/ab11
```

6. If staging succeeds, stop here.
7. If staging failed again, SAM-QFS again sets the damaged flag. Note the major inode number in the output of the `sls -D` command for the damaged copy.

In the example, the inode number of the file, `/samqfs1/genfiles/ab11`, is 1460:

```
root@solaris:~# sls -D /samqfs1/genfiles/ab11
/samqfs1/genfiles/ab11:
mode: -rw-r----- links: 1 owner: root group: other
length: 380051 admin id: 0 inode: 1460.1
project: system(0)
copy 1: ---D Mar 01 10:21 431.21bc6 li VOL024
...
root@solaris:~#
```

8. Look for possible causes. First, examine the SAM-QFS `/var/adm/sam-log` file for staging related messages that pertain to the inode of the file with the damaged copy.

The search can be carried out in various ways. In the example, we list the contents of log file using the Solaris `cat` command and pipe the output to `grep` and a regular expression that matches the inode number. We find two messages. Both indicate an I/O error and one explicitly implicates equipment (eq) ordinal number 804, one of our tape drives:

```
root@solaris:~# cat /var/adm/sam-log | grep "inode 1460"
Mar 11 15:35:44 server1 genu-20[8899]: Stage request canceled for inode 1460
(eq 804): I/O error.
Jan 11 15:35:44 server1 samfs[8894]: /sam4 inode 1460.1 - Archive copy 1 marked
damaged: I/O error
```

9. If the `/var/adm/sam-log` file implicates a specific SAM-QFS equipment ordinal number, examine the device log,
`/var/opt/SUNWsamfs/devlog/drive-equipment-number`, where *drive-equipment-number* is the ordinal number listed in the `/var/adm/sam-log` file.
10. If the problem appears to be specific to a particular drive, make the implicated drive unavailable to the staging process using the command `samcmd unavail drive-equipment-number`. Then undamage the copy, and try to stage it again.

```
root@solaris:~# samcmd unavail 804
```

```
root@solaris:~# stage -c 1 -I /samqfs1/genfiles/ab11
root@solaris:~# undamage -c1 /samqfs1/genfiles/ab11
root@solaris:~#
```

11. If staging fails again or if no single drive appears to be at fault, try to recover the copy using the `request` and `star` commands, as described in ["Recovering Files Using Archiver Log Entries"](#) on page 5-2, or Solaris utilities such as `tar` and `dd`.
12. If you still cannot recover the file and if the value of the data warrants it, engage a data recovery service. For assistance with Oracle StorageTek tape media, engage Oracle StorageTek Enterprise Tape Data Recovery services. Log in to My Oracle Support at support.oracle.com. Open a Service Request, select the tape drive model from the list under the request category, and select Media Issues from the list under subcategory.
13. If the file proves to be irrecoverable, unarchive the damaged copy. Use the command `unarchive -c CopyNumber file`, where *CopyNumber* is an integer representing the copy number and *file* is the path and file name of the damaged file.

```
root@solaris:~# unarchive -c 1 /samqfs1/genfiles/ab11
root@solaris:~#
```

14. Resolve any drive or media issues that are revealed by the log files.
15. If you disabled archiving, staging, and recycling processes in a previous step, re-enable them now. Go to ["Restoring Archiving File Systems to Normal Operation"](#) on page 6-1.
16. Otherwise, stop here.

Once you have completed configuration and data recovery, you should perform two last tasks:

- [Restoring Archiving File Systems to Normal Operation](#)
- [Preserving the New Configuration Information.](#)

Restoring Archiving File Systems to Normal Operation

If you disabled archiving and recycling, re-enable them now:

- [Enable Archiving](#)
- [Enable Recycling.](#)

Enable Archiving

1. Log in to the file-system metadata server as root.

```
root@solaris:~#
```

2. Open the `/etc/opt/SUNWsamfs/archiver.cmd` file in a text editor, and scroll down to the first `wait` directive that you added to the file when you started recovery efforts.

In the example, we use the `vi` editor:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# Configuration file for SAM-QFS archiving file systems
#-----
# General Directives
archivemeta = off
examine = noscan
#-----
# Archive Set Assignments
wait
fs = samqfs1
logfile = /var/adm/samqfs1.archive.log
all .
    1 -norelease 15m
    2 -norelease 15m
fs = samqfs2
logfile = /var/adm/samqfs2.archive.log
all .
...
```

3. To enable archiving, delete every `wait` directive that you added to the file when you started recovery efforts. Save the file, and close the editor.

In the example, we remove the single `wait` directive that we added:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#-----
# Archive Set Assignments
fs = samqfs1
logfile = /var/adm/samqfs1.archive.log
all .
    1 -norelease 15m
    2 -norelease 15m
    3 -norelease 15m
fs = samqfs2
...
:wq
root@solaris:~#
```

4. Next, [Enable Recycling](#).

Enable Recycling

1. Log in to the file-system metadata server as `root`.

```
root@solaris:~#
```

2. Open the `/etc/opt/SUNWsamfs/recycler.cmd` file in a text editor, and scroll down to the first `-ignore` parameter that you added to the file when you started recovery efforts.

In the example, we use the `vi` editor:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for SAM-QFS archiving file systems
#-----
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
library1 -hwm 95 -mingain 60 -ignore
```

3. Remove every `-ignore` parameter that you added when you started recovery efforts. Then save the file, and close the editor.

In the example, we have only one library in the SAM-QFS configuration, `library1`:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for SAM-QFS archiving file systems
#-----
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
library1 -hwm 95 -mingain 60 -ignore
:wq
root@solaris:~#
```

4. Check the modified configuration files for errors. Use the commands `archiver -lv` to check the `archiver.cmd` file and run the initialization command `sam-fsd`. Correct any errors.

In the example, the configuration files are correct:

```
root@solaris:~# archiver -lv
```

```

Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
1: #-----
2: # General Directives
3: archivemeta = off
4: examine = noscan
5: #-----
5: # Archive Set Assignments
7: fs = samqfs1
...
.sort: path
root@solaris:~# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
root@solaris:~#

```

5. Reconfigure the SAM-QFS software using the restored configuration files. Use the command `samd config`.

Archiving and recycling processes resume.

```
root@solaris:~# samd config
```

6. If you are recovering from a server problem or from loss or damage to one more file systems, [Save the Newly Restored SAM-QFS Configuration](#).
7. Otherwise, stop here.

Preserving the New Configuration Information

If you have changed the SAM-QFS configuration in the course of recovery efforts, you should back up the configuration again now.

Save the Newly Restored SAM-QFS Configuration

1. Log in to the file-system metadata server as root.

```
root@solaris:~#
```

2. Run the `samexplorer` command and create a SAMreport. Save it in the directory that holds your backup configuration information. Use the command `samexplorer path/hostname.YYYYMMDD.hhmmz.tar.gz`, where *path* is the path to the chosen directory, *hostname* is the name of the SAM-QFS file system host, and *YYYYMMDD.hhmmz* is a date and time stamp.

The default file name is `/tmp/SAMreport.hostname.YYYYMMDD.hhmmz.tar.gz`. In the example, we already have a directory for saving SAMreports, `/zfs1/sam_config/`. So we create the report in this directory:

```

root@solaris:~# samexplorer /zfs1/sam_
config/explorer/server1.20140430.1659MST.tar.gz
Report name:      /zfs1/sam_
config/explorer/samhost1.20140430.1659MST.tar.gz
Lines per file:   1000
Output format:    tar.gz (default) Use -u for unarchived/uncompressed.

Please wait.....

```

```
Please wait.....
Please wait.....
```

The following files should now be ftp'ed to your support provider as ftp type binary.

```
/zfs1/sam_config/explorer/samhost1.20140430.1659MST.tar.gz
```

3. Copy the configuration files to another file system.

```
/etc/opt/SUNWsamfs/
mcf
archiver.cmd
defaults.conf
diskvols.conf
hosts.family-set-name
hosts.family-set-name.local
preview.cmd
recycler.cmd
releaser.cmd
rft.cmd
samfs.cmd
stager.cmd
inquiry.conf
samremote                # SAM-Remote server configuration file
family-set-name           # SAM-Remote client configuration file
network-attached-library # Parameters file
scripts/*                 # Back up all locally modified files
/var/opt/SUNWsamfs/
```

4. Back up all library catalog data, including that maintained by the historian. For each catalog, use the command `/opt/SUNWsamfs/sbin/dump_cat -V catalog-file`, where *catalog-file* is the path and name of the catalog file. Redirect the output to *dump-file*, in a new location.

In the example, we dump the catalog data for `library1` to the file `library1cat.dump` in a directory on the independent NFS-mounted file system `zfs1`:

```
root@solaris:~# dump_cat -V /var/opt/SUNWsamfs/catalog/library1cat > \
/zfs1/sam_config/20140513/catalogs/library1cat.dump
```

5. Copy system configuration files that were modified during SAM-QFS installation and configuration. These may include:

```
/etc/
syslog.conf
system
vfstab
/kernel/drv/
sgen.conf
samst.conf
samrd.conf
sd.conf
ssd.conf
st.conf
/usr/kernel/drv/dst.conf
```

6. Copy any custom shell scripts and `crontab` entries that you created as part of the SAM-QFS configuration to the selected subdirectory.

For example, if you created a `crontab` entry to manage creation of recovery points, you would save a copy now.

7. Record the revision level of the currently installed software, including Oracle SAM-QFS, Solaris, and Solaris Cluster (if applicable), and save a copy of the information in a `readme` file in the chosen subdirectory.
8. In the chosen subdirectory, save copies of any newly downloaded Oracle SAM-QFS, Solaris, and Solaris Cluster packages so that you can restore the software quickly, should it again become necessary.

Understanding the Archiver Log

The following table defines each field in the archiver log.

Field	Typical Value	Meaning
1	A	Archive activity: A (A rchived), R (R e-archived), or U (U narchived)
2	2014/03/23	Date of the archive action, in <i>yyyy/mm/dd</i> format.
3	18:42:06	Time of the archive activity, in <i>hh:mm:ss</i> format.
4	mo	Archive media type. For information about media types, see mcf(4) in SAM-QFS and Sun Storage Archive Manager Reference Manual.
5	0004A	VSN. For removable media cartridges, the volume serial name. For disk archives, the disk volume name and archive tar(1) file path.
6	arset0.1	Archive set and copy number.
7	9a089.1329	Physical position of the start of the archive file on media (tar file) and file offset within the archive file, in hexadecimal format.
8	samfs1	File system name.
9	118.51	Inode number and generation number. The generation number is used in addition to the inode number for uniqueness because inode numbers are reused.
10	162514	Length of the file if the file is written on only one volume. Length of the section if the file is written on multiple volumes.
11	t0/fdn	Path and name of the file relative to the file system's mount point.
12	f	Type of file: d (d irectory), f (f ile), l (symbolic l ink), R (R emovable-media file), I (segment I ndex), or S (data S egment)
13	0	Section of an overflowed file or segment. If the file is an overflowed file, the value is nonzero. For all other file types, the value is 0.
14	56	Equipment ordinal of the drive on which the file was archived.

The following example shows sample lines from an archiver log file.

```
A 2014/03/23 18:42:06 mo 0004A arset0.1 9a089.1329 samfs1 118.51 162514 t0/fdn f 0
56
A 2014/03/23 18:42:10 mo 0004A arset0.1 9aac2.1 samfs1 189.53 1515016 t0/fae f 0
56
A 2014/03/23 18:42:10 mo 0004A arset0.1 9aac2.b92 samfs1 125.53 867101 t0/fai f 0
56
A 2014/03/23 19:13:09 lt SLOT22 arset0.2 798.1 samfs1 71531.14 1841087 t0/fhh f 0
51
A 2014/03/23 19:13:10 lt SLOT22 arset0.2 798.e0e samfs1 71532.12 543390 t0/fhg f 0
```

51

```
A 2014/03/24 13:30:24 dk DISK01/d8/d16/f216 arset4.1 810d8.1 qfs2 119571.301
1136048 t1/fileem f 0 0
A 2014/03/24 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.8ad qfs2 119573.295
1849474
t1/fileud f 0 0
A 2014/03/24 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.16cb qfs2 119576.301
644930 t1/fileen f 0 0
A 2014/03/24 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.1bb8 qfs2 119577.301
1322899 t1/fileeo f 0 0
```

Glossary of Equipment Types

The value of the `Equipment Type` field of the Master Configuration File (`mcf`) identifies devices and device configurations within the SAM-QFS software using two-character codes. This glossary lists the codes for quick reference when working with the samples or when interpreting an existing `mcf` (for full details see the `mcf(4)` man page).

For convenience, the codes are divided into three sections and then listed alphabetically:

- [Recommended Equipment and Media Types](#)
- [Other Equipment and Media Types](#)

Recommended Equipment and Media Types

This section describes all of the equipment codes that you normally need: the generic equipment codes (`rb`, `tp`, and `od`) and codes for identifying network-attached library interfaces and the SAM-QFS historian.

The generic equipment codes `rb`, `tp`, and `od` are the preferred equipment type codes for all SCSI-attached libraries, tape drives, and optical disk devices. When you specify a generic equipment type, SAM-QFS can automatically set the correct type based on SCSI vendor codes.

gXXX

Where `XXX` is an integer in the range [0-127], a striped group of disk devices that is part of an `ma` disk-cache family set.

hy

The SAM-QFS historian, an optional, virtual library that maintains a media catalog, but has no associated hardware. Used for tracking exported media.

ma

A high-performance QFS file system that maintains file-system metadata on one or more dedicated `mm` disk devices. File data resides on separate `md`, `mr`, or `gXXX` data devices.

md

A disk device that stores file data for an `ma` file system or data and metadata for an `ms` file system. `md` devices store file data in small, 4-kilobyte Disk Allocation Units (DAUs) and large, 16-, 32-, or 64-kilobyte DAUs. The default DAU is 64-kilobytes.

mm

A disk device that stores file-system metadata for a high-performance `ma` file system.

mr

A disk device that stores file data for an `ma` file system. `mr` devices store file data in large, fully adjustable Disk Allocation Units (DAUs) that are multiples of 8 kilobytes in the range 8-65528 kilobytes. The default DAU is 64 kilobytes.

ms

A SAM-QFS file system that maintains file-system metadata on the same devices that store file data.

od

Any SCSI-attached optical disk. SAM-QFS sets the appropriate equipment type automatically using the SCSI vendor code.

rb

Any SCSI-attached tape library. SAM-QFS sets the appropriate equipment type automatically using the SCSI vendor code.

rd

The SAM-Remote pseudo-device. In the Master Configuration File (`mcf`), the corresponding `Equipment Identifier` field has to contain the path to the pseudo-device (such as `/dev/samrd/rd2`). The corresponding `Family Set` field has to contain the hostname of the SAM-Remote server.

sc

A SAM-Remote client system. In the Master Configuration File (`mcf`), the corresponding `Equipment Identifier` field has to contain the path the SAM-Remote client-configuration file for the client. The corresponding `Family Set` field has to contain the family set name of the server. The `Additional Parameters` field must contain the full path to the client's library catalog file.

sk

An Oracle StorageTek ACSLS interface to a network-attached library. In the Master Configuration File (`mcf`), the corresponding `Equipment Identifier` field has to contain the path to the parameters file for the ACSLS interface. For more information, see the `stk(7)` man page.

ss

A SAM-Remote server. In the Master Configuration File (`mcf`), the corresponding `Equipment Identifier` field has to contain the path to the SAM-Remote server-configuration file. The corresponding `Family Set` field has to contain the family set name of the server, which must match the name used in the `Family Set` field of the `mcf` on the client.

tp

Any SCSI-attached tape drive. SAM-QFS sets the appropriate equipment type automatically using the SCSI vendor code.

Other Equipment and Media Types

The equipment types listed in this section are also supported.

Note that, in most cases, Oracle recommends identifying SCSI-attached libraries, tape drives, and optical disk devices using the generic equipment types `rb`, `tp`, and `od`. The generic equipment types tell SAM-QFS to identify the hardware dynamically, using SCSI vendor IDs. The type codes below are essential when migrating from one media type to another and may sometimes be useful for management purposes. But using

them in a Master Configuration File (`mcf`), for example, hard-codes a static equipment configuration that may not match the actual hardware.

ac

A Sun 1800, 3500, or L11000 tape library.

at

A Sony AIT-4 or AIT-5 tape drive.

cy

A Cygnet optical disk library.

d3

A StorageTek D3 tape drive.

dm

A Sony DMF library.

ds

A DocuStore or Plasmon optical disk library.

dt

A DAT 4-mm tape drive.

e8

An Exabyte X80 library.

fd

A Fujitsu M8100 128-track tape drive.

h4

An HP SL48 or SL24 library.

hc

An Hewlett Packard L9-/L20-/L60-series library.

i7

An IBM 3570 tape drive.

ic

An IBM 3570 media changer.

il

An IBM 3584 tape library.

li

An LTO-3 or later tape drive.

lt

A Digital Linear Tape (DLT), Super DLT, or DLT-S4 tape drive.

me

A Metrum library.

mf

An IBM Multi Function optical drive.

mo

A 5.25-in erasable optical drive.

o2

A 12-in WORM drive.

ov

An Overland Data Inc. Neo Series tape library.

pd

A Plasmon D-Series DVD-RAM library.

q8

A Qualstar 42xx, 62xx, or 82xx library.

s3

A StorageTek SL3000 library.

s9

An Oracle StorageTek 97xx series library.

se

A StorageTek 9490 tape drive.

sf

A StorageTek T9940 tape drive.

sg

A StorageTek 9840C or later tape drive.

sl

A Spectra Logic or Qualstar tape library.

st

A StorageTek 3480 tape drive.

ti

A StorageTek T10000 (Titanium) tape drive.

vt

A Metrum VHS (RSP-2150) tape drive.

wo

A 5.25-in optical WORM drive.

xt

An Exabyte (850x) 8-mm tape drive.

Product Accessibility Features

Users with low vision, blindness, color blindness, or other visual impairments can access the StorageTek Storage Archive Manager and StorageTek QFS Software (SAM-QFS) via the commandline interface. This text-based interface is compatible with screen readers, and all functions are controlled using a keyboard.

Glossary

This glossary focuses on terms specific to SAM-QFS software and file systems. For industry standard definitions, please refer to the dictionary maintained by the Storage Networking Industry Association at <http://www.snia.org/education/dictionary/>.

addressable storage

The storage space encompassing online, nearline, offsite, and offline storage that is user-referenced through a SAM-QFS file system.

admin set ID

A storage administrator-defined set of users and/or groups that share common characteristics. Admin sets are typically created to administer storage for projects that involve users from several groups and span multiple files and directories.

archival media

The media to which an archive file is written. Archival media includes both removable tape or magneto-optical cartridges and disk file systems configured for archiving.

archival storage

Data storage space created on archival media.

archive set

An archive set identifies a group of files to be archived, and the files share common criteria that pertain to the size, ownership, group, or directory location. Archive sets can be defined across any group of file systems.

archiver

The archive program that automatically controls the copying of files to removable cartridges.

associative staging

Staging a group of related files when any one member of the group is staged. When files inhabit the same directory and are frequently used together, file owners can associate them by setting the SAM-QFS associative-staging file attribute. Then if any files in the group are offline when one of them is accessed by an application, SAM-QFS stages the entire group from archival media to disk cache. This insures that all needed files re available at the same time.

audit (full)

The process of loading cartridges to verify their VSNs. For magneto-optical cartridges, the capacity and space information is determined and entered into the automated library's catalog. See [volume serial number \(VSN\)](#).

automated library

A robotically controlled device designed to automatically load and unload removable media cartridges without operator intervention. An automated library contains one or more drives and a transport mechanism that moves cartridges to and from the storage slots and the drives.

backup

A snapshot of a collection of files for the purpose of preventing inadvertent loss. A backup includes both the file's attributes and associated data.

block allocation map

A bitmap representing each available block of storage on a disk and indicating whether the block is in use or free.

block size

The size of the smallest addressable data unit on a block device, such as a hard disk or magnetic tape cartridge. On disk devices, this is equivalent to the *sector size*, which is typically 512 bytes.

cartridge

A container for data-storage media, such as magnetic tape or optical media. Also called a [volume](#), *tape*, or *piece of media*. See [volume serial number \(VSN\)](#).

catalog

A record of the removable media volumes in an automated library. There is one catalog for each automated library and, at a site, there is one historian for all automated libraries. Volumes are identified and tracked using a [volume serial number \(VSN\)](#).

client-server

The model of interaction in a distributed system in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called the client. The program satisfying the response is called the server.

connection

The path between two protocol modules that provides reliable stream delivery service. A TCP connection extends from a TCP module on one machine to a TCP module on the other.

data device

In a file system, a device or group of devices upon which file data is stored.

DAU

See [disk allocation unit \(DAU\)](#).

device logging

A configurable feature that provides specific error information for the hardware devices that support a SAM-QFS file system.

device scanner

Software that periodically monitors the presence of all manually mounted removable devices and that detects the presence of mounted cartridges that can be requested by a user or other process.

direct access

A file attribute (stage never) designating that a nearline file can be accessed directly from the archive media and need not be retrieved to disk cache.

direct attached library

An automated library connected directly to a server using a SCSI interface. A SCSI-attached library is controlled directly by the SAM-QFS software.

direct I/O

An attribute used for large block-aligned sequential I/O. The `setfa` command's `-D` option is the direct I/O option. It sets the direct I/O attribute for a file or directory. If applied to a directory, the direct I/O attribute is inherited.

directory

A file data structure that points to other files and directories within the file system.

disk allocation unit (DAU)

In SAM-QFS file systems, the minimum amount of contiguous space that each I/O operation consumes, regardless of the amount of data written. The disk allocation unit thus determines minimum number of I/O operations needed when transferring a file of a given size. It should be a multiple of the [block size](#) of the disk device.

Disk allocation unit vary depending upon the SAM-QFS device type selected and user requirements. The `md` device type uses dual-allocation units: the DAU is 4 kilobytes for the first eight writes to a file and then a user-specified 16, 32, or 64 kilobytes for any subsequent writes, so that small files are written in suitably small blocks, while larger files are written in larger blocks. The `mr` and [striped group](#) device types use a DAU that is adjustable in increments of 8 within the range [8-65528] kilobytes. Files are thus written in large, uniform blocks that can closely approximate the size of the large, uniformly sized files.

disk buffer

In a SAM-Remote configuration, the buffer on the server system that is used for archiving data from the client to the server.

disk cache

The disk-resident portion of the file system software, used to create and manage data files between online disk cache and archive media. Individual disk partitions or an entire disk can be used as disk cache.

disk space threshold

The maximum or minimum level of disk cache utilization, as defined by an administrator. The releaser controls disk cache utilization based on these predefined disk space thresholds.

disk striping

The process of recording a file across several disks, thereby improving access performance and increasing overall storage capacity. See also [striping](#).

drive

A mechanism for transferring data to and from a removable media volume.

Ethernet

A packet-switched local-area network technology.

extent array

The array within a file's inode that defines the disk location of each data block assigned to the file.

family device set

See [family set](#).

family set

A storage device that is represented by a group of independent physical devices, such as a collection of disks or the drives within an automated library. See also [storage family set](#).

FDDI

Fiber-distributed data interface, a standard for data transmission in a local area network that can extend in range up to 200 km (124 miles). The FDDI protocol is based on the token ring protocol.

Fibre Channel

The ANSI standard that specifies high-speed serial communication between devices. Fibre Channel is used as one of the bus architectures in SCSI-3.

file system

A hierarchical collection of files and directories.

file-system-specific directives

Archiver and releaser directives that follow global directives in the `archiver.cmd` file, are specific to a particular file system, and begin with `fs =`. File-system-specific directives apply until the next `fs =` directive line or the end of file is encountered. If multiple directives affect a file system, the file-system-specific directives override the global directives.

ftp

File Transfer Protocol, a network protocol for transferring files between two hosts. For a more secure alternative, see [sftp](#).

global directives

Archiver and releaser directives that apply to all file systems and that appear before the first `fs=` line.

grace period

In a [quota](#), the amount of time that the file system allows the total size of files belonging to specified user, group, and/or [admin set IDs](#) to exceed the [soft limit](#) specified in the quota.

hard limit

In a [quota](#), the absolute maximum quantity of storage resources that specified user, group, and/or [admin set IDs](#) can consume. See [soft limit](#).

high-water mark

1. In an archiving file system, the percentage disk-cache utilization at which SAM-QFS file systems start the releaser process, deleting previously archived files from disk. A properly configured high-water mark insures that the file system always has enough space available for new and newly staged files. For more information, see the `sam-releaser` and `mount_samfs` man pages. Compare [low-water mark](#).
2. In a removable media library that is part of an archiving file system, the percentage media-cache utilization that starts the recycler process. Recycling empties partially full volumes of current data so that they can be replaced by new media or relabeled.

historian

The SAM-QFS historian is a catalog of volumes that have been exported from automated media libraries that are defined in the `/etc/opt/SUNWsamfs/mcf` file. By default, it is located on the SAM-QFS file-system host at `/var/opt/SUNWsamfs/catalog/historian`. For details, see the SAM-QFS historian man page.

hosts file

The hosts file contains a list of all of the hosts in a shared file system. If you are initializing a file system as a SAM-QFS shared file system, the hosts file, `/etc/opt/SUNWsamfs/hosts.fs-name`, must be created before the file system is created. The `sammkfs` command uses the hosts file when it creates the file system. You can use the `samsharefs` command to replace or update the contents of the hosts file at a later date.

indirect block

A disk block that contains a list of storage blocks. File systems have up to three levels of indirect blocks. A first-level indirect block contains a list of blocks used for data storage. A second-level indirect block contains a list of first-level indirect blocks. A third-level indirect block contains a list of second-level indirect blocks.

inode

Index node. A data structure used by the file system to describe a file. An inode describes all the attributes associated with a file other than the name. The attributes include ownership, access, permission, size, and the file location on the disk system.

inode file

A special file (`.inodes`) on the file system that contains the inode structures for all files resident in the file system. Inodes are 512 bytes long. The inode file is a metadata file, which is separated from file data in the file system.

kernel

The program that provides basic operating system facilities. The UNIX kernel creates and manages processes, provides functions to access the file system, provides general security, and supplies communication facilities.

LAN

Local area network.

lease

A function that grants a client host permission to perform an operation on a file for a specified period of time. The metadata server issues leases to each client host. The leases are renewed as necessary to permit continued file operations.

library

See [automated library](#).

library catalog

See [catalog](#).

local file system

A file system that is installed on one node of a Solaris Cluster system and is not made highly available to another node. Also, a file system that is installed on a server.

low-water mark

In an archiving file system, the percentage disk-cache utilization at which SAM-QFS file systems stops the releaser process and stops deleting previously archived files from disk. A properly configured low-water mark insures that the file system retains as many file in cache as possible, for best performance, while making space available for new and newly staged files. For more information, see the `sam-releaser` and `mount_samfs` man pages. Compare [high-water mark](#).

LUN

Logical unit number.

mcf

Master Configuration File. The file that is read at initialization time that defines the relationships between the devices (the topology) in a file system environment.

media

Tape or optical disk cartridges.

media recycling

The process of recycling or reusing archive media with few active files.

metadata

Data about data. Metadata is the index information used to locate the exact data position of a file on a disk. It consists of information about files, directories, access control lists, symbolic links, removable media, segmented files, and the indexes of segmented files.

metadata device

A device (for example, a solid-state disk or mirrored device) upon which file system metadata is stored. Having file data and metadata on separate devices can increase performance. In the `mcf` file, a metadata device is declared as an `mm` device within an `ma` file system.

mirror writing

The process of maintaining two copies of a file on disjointed sets of disks to prevent loss from a single disk failure.

mount point

The directory on which a file system is mounted.

multireader file system

A single-writer, multireader capability that enables you to specify a file system that can be mounted on multiple hosts. Multiple hosts can read the file system, but only one host can write to the file system. Multiple readers are specified with the `-o reader` option with the `mount` command. The single-writer host is specified with the `-o writer` option with the `mount` command. For more information, see the `mount_samfs` man page.

name space

The metadata portion of a collection of files that identifies the file, its attributes, and its storage locations.

nearline storage

Removable media storage that requires robotic mounting before it can be accessed. Nearline storage is usually less expensive than online storage, but it takes somewhat longer to access.

network attached automated library

A library, such as those from StorageTek, ADIC/Grau, IBM, or Sony, that is controlled using a software package supplied by the vendor. The QFS file system interfaces with the vendor software using a SAM-QFS media changer daemon designed specifically for the automated library.

NFS

Network file system, a file system that provides transparent access to remote file systems on heterogeneous networks.

NIS

Network Information Service, a distributed network database containing key information about systems and users on the network. The NIS database is stored on the master server and all slave servers.

offline storage

Storage that requires operator intervention for loading.

offsite storage

Storage that is remote from the server and is used for disaster recovery.

online storage

Storage that is immediately available, such as disk cache storage.

partition

A portion of a device or a side of a magneto-optical cartridge.

preallocation

The process of reserving a contiguous amount of space on the disk cache for writing a file. Preallocation can be specified only for a file that is size zero. For more information, see the `setfa` man page.

pseudo device

A software subsystem or driver with no associated hardware.

QFS

The SAM-QFS QFS Software product, a high-performance, high-capacity, UNIX file system that can be used on its own or as an archiving file system controlled by SAM-QFS Storage Archive Manager.

qfsdump

See [samfsdump \(qfsdump\)](#).

qfsrestore

See [samfsrestore \(qfsrestore\)](#).

quota

The amount of storage resources that specified user, group, or [admin set IDs](#) are allowed to consume. See [hard limit](#) and [soft limit](#).

RAID

Redundant array of independent disks. A disk technology that uses several independent disks to reliably store files. It can protect against data loss from a single disk failure, can provide a fault-tolerant disk environment, and can provide higher throughput than individual disks.

recovery point

A compressed file that stores a point-in-time backup copy of the metadata for a SAM-QFS file system.

In the event of a data loss—anything from accidental deletion of a user file to catastrophic loss of a whole file system—an administrator can recover to the last known-good state of the file or file system almost immediately by locating the last recovery point at which the file or file system remained intact. The administrator then restores the metadata recorded at that time and either stages the files indicated in the metadata to the disk cache from archival media or, preferably, lets the file system stage files on demand, as users and applications access them.

recycler

A SAM-QFS utility that reclaims space on cartridges that is occupied by expired archive copies.

regular expression

A string of characters in a standardized pattern-matching language that is designed for searching, selecting, and editing other character strings, such as file names and configuration files. For full details of the regular expression syntax used in SAM-QFS file-system operations, see the SAM-QFS Solaris [regex](#) and [regcmp](#) man pages.

release priority

The priority according to which a file in a file system is released after being archived. Release priority is calculated by multiplication of various weights of file properties and then summation of the results.

releaser

A SAM-QFS component that identifies archived files and releases their disk cache copies, thus making more disk cache space available. The releaser automatically regulates the amount of online disk storage according to high and low thresholds.

remote procedure call

See [RPC](#).

removable media file

A special type of user file that can be accessed directly from where it resides on a removable media cartridge, such as magnetic tape or optical disk cartridge. Also used for writing archive and stage file data.

robot

An [automated library](#) component that moves cartridges between storage slots and drives. Also called a [transport](#).

round-robin

A data access method in which entire files are written to logical disks in a sequential fashion. When a single file is written to disk, the entire file is written to the first logical disk. The second file is written to the next logical disk, and so on. The size of each file determines the size of the I/O. See also [disk striping](#) and [striping](#).

RPC

Remote procedure call. The underlying data exchange mechanism used by NFS to implement custom network data servers.

SAM

See [SAM-QFS](#).

SAM-Remote client

A SAM-QFS system with a client daemon that contains a number of pseudodevices, and can also have its own library devices. The client depends on a SAM-Remote server for archive media for one or more archive copies.

SAM-Remote server

Both a full-capacity SAM-QFS storage management server and a SAM-Remote server daemon that defines libraries to be shared among SAM-Remote clients.

SAM-QFS

1. A common abbreviation for the SAM-QFS product.
2. An adjective describing a [QFS](#) file system that is configured for archiving and managed by SAM-QFS Storage Archive Manager software.

samfsdump (qfsdump)

A program that creates a control structure dump and copies all the control structure information for a given group of files. It does not generally copy file data. With the `-u` option, the command also copies data files. If the Storage Archive Manager packages are not installed, the command is called `qfsdump`.

samfsrestore (qfsrestore)

A program that restores inode and directory information from a control structure dump. See also [samfsdump \(qfsdump\)](#).

SAN

Storage Area Network.

SCSI

Small Computer System Interface, an electrical communication specification commonly used for peripheral devices such as disk and tape drives and automated libraries.

shared hosts file

When you create a shared file system, the system copies information from the hosts file to the shared hosts file on the metadata server. You update this information when you issue the **samsharefs -u** command

Small Computer System Interface

See [SCSI](#).

soft limit

In a [quota](#), the maximum amount of storage space that a specified user, group, and/or [admin set IDs](#) can fill for an indefinite period. Files can use more space than the soft limit allows, up to the hard limit, but only for a short [grace period](#) defined in the quota. See [hard limit](#).

sftp

Secure File Transfer Protocol, a secure implementation of [ftp](#) based on [ssh](#).

ssh

Secure Shell, an encrypted network protocol that allows secure, remote command-line login and command execution.

staging

The process of copying a nearline or offline file from archive storage back to online storage.

storage family set

A set of disks that are collectively represented by a single disk family device.

storage slots

Locations inside an automated library in which cartridges are stored when not being used in a drive.

stripe size

The number of disk allocation units (DAUs) to be allocated before writing proceeds to the next device of a stripe. If the `stripe=0` mount option is used, the file system uses round-robin access, not striped access.

striped group

A collection of devices within a file system that is defined in the `mcf` file as one or more `gxxx` devices. Striped groups are treated as one logical device and are always striped with a size equal to the disk allocation unit (DAU).

striping

A data access method in which files are simultaneously written to logical disks in an interlaced fashion. SAM-QFS file systems provide two types of striping: "hard

striping," using stripe groups, and "soft striping," using the `stripe=x` mount parameter. Hard striping is enabled when a file system is set up, and requires the definition of stripe groups within the `mcf` file. Soft striping is enabled through the `stripe=x` mount parameter, and can be changed for the file system or for individual files. It is disabled by setting `stripe=0`. Hard and soft striping can both be used if a file system is composed of multiple stripe groups with the same number of elements. See also [round-robin](#).

SUNW.qfs

A Solaris Cluster resource type that supports SAM-QFS shared file systems. The `SUNW.qfs` resource type defines failover resources for the shared file system's metadata server (MDS)

superblock

A data structure in the file system that defines the basic parameters of the file system. The superblock is written to all partitions in the storage family set and identifies the partition's membership in the set.

tar

Tape archive. A standard file and data recording format used for archive images.

TCP/IP

Transmission Control Protocol/Internet Protocol. The internet protocols responsible for host-to-host addressing and routing, packet delivery (IP), and reliable delivery of data between application points (TCP).

timer

Quota software that keeps track of the period starting when a user reaches a soft limit and ending when the hard limit is imposed on the user.

transport

See [robot](#).

vfstab file

The `vfstab` file contains mount options for the file system. Mount options specified on the command line override those specified in the `/etc/vfstab` file, but mount options specified in the `/etc/vfstab` file override those specified in the `samfs.cmd` file.

volume

1. On storage media, a single, accessible, logical storage area, usually addressed by a [volume serial number \(VSN\)](#) and/or volume label. Storage disks and magnetic tape cartridges can hold one or more volumes. For use, volumes are *mounted* on a file system at a specified [mount point](#).
2. A magnetic tape [cartridge](#) that holds a single logical volume.
3. On a random-access disk device, a file system, directory or file that is configured and used as if it were a sequential-access, removable-media cartridge, such as a tape.

volume overflow

A capability that enables the system to span a single file over multiple [volumes](#). Volume overflow is useful for sites using very large files that exceed the capacity of their individual cartridges.

volume serial number (VSN)

1. A serial number assigned to a tape or disk storage volume. A volume serial number can consist of up to six uppercase, alphanumeric characters, must start with a letter, and must identify the volume uniquely within a given context, such a tape library or partition. The volume serial number is written on the volume label.
2. Loosely, a specific storage **volume**, especially a removable media **cartridge**.

WORM

Write-Once-Read-Many. A storage classification for media that can be written only once but read many times.